The Round Complexity of Perfectly Secure General VSS

Ashish Choudhury[1] ⋆, Kaoru Kurosawa[2], and Arpita Patra[3] ⋆⋆

[1] Applied Statistics Unit
Indian Statistical Institute Kolkata India
partho_31@yahoo.co.in, partho31@gmail.com
[2] Department of Computer and Information Sciences
Ibaraki University, Hitachi Ibaraki, Japan
kurosawa@mx.ibaraki.ac.jp
[3] Department of Computer Science
Aarhus University Denmark
arpitapatra10@gmail.com, arpita@cs.au.dk

Abstract. The round complexity of verifiable secret sharing (VSS) schemes has been studied extensively for threshold adversaries. In particular, Fitzi et al. showed an efficient 3-round VSS for $n \geq 3t + 1$ [4], where an infinitely powerful adversary can corrupt t (or less) parties out of n parties. This paper shows that for non-threshold adversaries:

1. Two round perfectly secure VSS is possible if and only if the underlying adversary structure satisfies the $\mathcal{Q}^4$ condition;
2. Three round perfectly secure VSS is possible if and only if the underlying adversary structure satisfies the $\mathcal{Q}^3$ condition.

Further as a special case of our three round protocol, we can obtain a more efficient 3-round VSS than the VSS of Fitzi et al. for $n = 3t + 1$. More precisely, the communication complexity of the reconstruction phase is reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$. We finally point out a flaw in the reconstruction phase of the VSS of Fitzi et al., and show how to fix it.

1 Introduction

Verifiable Secret Sharing (VSS) [2, 1] is a two phase (sharing, reconstruction) protocol, carried out among n parties and is used as a fundamental building block in many distributed cryptographic protocols. VSS extends the notion of secret sharing [10] to the *active* corruption model. In VSS protocols, an *infinitely powerful malicious adversary* can corrupt not only some subset of parties but also the *dealer*, who shares the secret. Even then, a unique secret is reconstructed in the reconstruction phase no matter how the malicious parties behave.

⋆ Financial Support from Department of Information Technology, Government of India Acknowledged.
⋆⋆ Financial Support from Center for Research in Foundations of Electronic Markets, Denmark Acknowledged.

Round complexity is one of the important complexity measures of any VSS protocol. Gennaro et al. [5] studied the round complexity of perfectly secure VSS, where they defined the round complexity of a VSS protocol as the number of *communication rounds* during the sharing phase. In their model, the n parties are pairwise connected by secure channels and a *common broadcast* channel is available, which allows any party to send some information identically to every other party. The adversary is characterized as a *threshold* adversary, who can corrupt any t parties. In such a model, Gennaro et al. showed the following:

1. Two round perfectly secure VSS is possible if and only if $n \geq 4t + 1$;
2. Three round perfectly secure VSS is possible if and only if $n \geq 3t + 1$.

Their 3-round VSS for $n \geq 3t + 1$ is *inefficient* while their 2-round VSS for $n \geq 4t + 1$ is efficient. A polynomial time, 3-round VSS for $n \geq 3t + 1$ was given by Fitzi et al. [4]. Later on, Katz et al. [7] improved the VSS of [4] in such a way that the broadcast channel is used for only one round during the sharing phase, whereas it is used for two rounds in [4].

1.1 Motivation of Our Work

Modeling the adversary by a threshold helps in easy characterization of the protocols and it also helps in analyzing the protocols. However, as mentioned in [6], modeling the (dis)trust in the network as a threshold adversary is not always appropriate because threshold protocol requires more *stringent* requirements than the reality. Let the set of n parties be denoted by $\mathcal{P} = \{P_1, \ldots, P_n\}$. Then a *non-threshold general adversary* $\mathcal{A}$ is characterized by an *adversary structure* Γ, which is a collection of the subsets of parties that the adversary $\mathcal{A}$ can *potentially* corrupt. That is, $\Gamma = \{B \subset \mathcal{P} \mid \mathcal{A} \text{ can corrupt } B\}$. Moreover, we assume that if $B \in \Gamma$ and if $B' \subset B$, then $B' \in \Gamma$.

Definition 1 ($\mathcal{Q}^k$ Condition [6]). *$\mathcal{A}$ satisfies $\mathcal{Q}^k$ condition with respect to $\mathcal{P}$, if there exists no k sets in Γ, which adds upto the whole set $\mathcal{P}$. That is:*

$$\forall B_1, \ldots, B_k \in \Gamma : B_1 \cup \ldots \cup B_k \neq \mathcal{P}.$$

Cramer et al. [3] showed a VSS for $\mathcal{Q}^3$ adversary structures by using a linear secret sharing scheme (LSSS). The VSS of [3] is *efficient* in the size of the underlying LSSS (see Sec. 2.2 for the definition of LSSS), but requires *more than seven rounds*. Maurer showed a *four round* VSS for $\mathcal{Q}^3$ adversary structures [9]. However, its computation and communication cost is *inefficient* [4].

In threshold settings, any $t + 1$ honest parties can reconstruct not only the secret s but also the randomness used by the dealer during the sharing phase. On the other hand, in non-threshold settings, an *access set* of parties can reconstruct *only* s, but not the randomness of the dealer in general. This is because the submatrix of the LSSS corresponding to an access set A is not necessarily of *full*

[4] We can see that its round complexity can be reduced to three by using the technique from [5] for making pairwise consistency checks. Still it is very inefficient.

rank (see Section 2 and in general [3] for more details). Due to this reason, a straightforward generalization of the techniques of [5, 4] will not work in non-threshold settings. Indeed [3] introduces a *commitment transfer protocol* and a *commitment sharing protocol* to design VSS for $\mathcal{Q}^3$ adversary structures [3].

Though there exist VSS protocols tolerating general adversary, to the best of our knowledge, *nothing is known in the literature regarding the round complexity of VSS tolerating general adversary.* This motivates us to do the same.

1.2 Our Results

We generalize the results of [5] and show the following:

1. Two round perfectly secure VSS is possible iff $\mathcal{A}$ satisfies the $\mathcal{Q}^4$ condition;
2. Three round perfectly secure VSS is possible iff $\mathcal{A}$ satisfies the $\mathcal{Q}^3$ condition.

In our 2-round VSS, the communication cost is polynomial in the size of the underlying LSSS, and the computation cost is polynomial in the size of Γ. So if $|\Gamma|$ and size of the underlying LSSS is polynomial then our 2-round scheme is efficient. On the other hand, in our 3-round VSS, both the communication and communication cost are polynomial in the size of the underlying LSSS. Thus if the underlying LSSS is polynomial then our 3-round scheme is efficient. Further as a special case of our 3-round protocol, we can obtain a more efficient 3-round VSS than the VSS of Fitzi et al. for $n = 3t+1$. More precisely, the communication complexity of the reconstruction phase is reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$.

Fitzi et al. [4] first designed a 3-round *weak secret sharing* (WSS) protocol. WSS is the same as VSS except for that a unique secret or $\perp$ must be reconstructed in the reconstruction phase (when the dealer is corrupted). Then they constructed their 3-round VSS by letting each party run the WSS as a dealer in parallel. Typically, a party participates in the reconstruction phase of his own WSS as like any other party and does not play any special role. On the other hand for constructing our VSS protocol, we first design a 3-round *weak commitment scheme* (WCS), and then replace the WSS with our WCS. An important difference now is that each party plays a special role in the reconstruction phase of his own WCS. It turns out that it is easier to construct a WCS than the WSS, and the efficiency is improved. Our WCS is also conceptually much simpler.

To design our 2-round VSS protocol, we generalize the techniques used in [5]. Notice that a straightforward generalization will not work, as the protocol of [5] uses the properties of Reed-Solomon codes [8]. To deal with this problem, we introduce the notion of $\mathcal{A}$-clique. Due to this, the resultant protocol performs computation which is polynomial in $|\Gamma|$. We finally point out a flaw in the reconstruction phase of the VSS of Fitzi et al., and show how to fix it.

2 Preliminaries

2.1 Secret Sharing Scheme

In a secret sharing scheme, a dealer $D \in \mathcal{P}$ distributes a secret $s \in \mathbb{F}$, where $\mathbb{F}$ is a finite field, to the parties in $\mathcal{P}$ in such a way that some subsets of the participants (called as access sets) can reconstruct s from their shares, while the other subsets of the participants (called forbidden sets) have no information about s from their shares. The family of access sets is called an *access structure*. Moreover, we assume that the access structure is *monotone* implying that if $A \in \Sigma$ and $A' \supseteq A$, then $A' \in \Sigma$. Corresponding to Σ, we have the *adversary structure* $\Gamma = \Sigma^c$, where c denotes the complement. The sets in Γ are called as forbidden sets. There exists a *computationally unbounded, adaptive, rushing* adversary $\mathcal{A}$, who can control any set in Γ. *However, it is assumed that D will not be under the control of $\mathcal{A}$ and every party under the control of $\mathcal{A}$ will follow the protocol instructions.*

2.2 Linear Secret Sharing Scheme (LSSS) [3]

A secret sharing scheme for any monotone access structure Σ can be realized by a LSSS [3] as follows: Let $\mathcal{M}$ be an $\ell \times e$ matrix over $\mathbb{F}$ and $\psi : \{1, \cdots, \ell\} \rightarrow \{1, \cdots, n\}$ be a labeling function, where $\ell \geq e$ and $\ell \geq n$.

Sharing algorithm:

1. To share a secret $s \in \mathbb{F}$, D first chooses a random vector $\boldsymbol{\rho} \in \mathbb{F}^{e-1}$ and compute a vector
$$\boldsymbol{v} = (v_1, \cdots, v_\ell)^T = \mathcal{M} \cdot \begin{pmatrix} s \\ \boldsymbol{\rho} \end{pmatrix}. \quad (1)$$
2. Let
$$\mathsf{LSSS}(s, \boldsymbol{\rho}) = (\texttt{share}_1, \cdots, \texttt{share}_n), \quad (2)$$
where $\texttt{share}_i = \{v_j \mid \psi(j) = i\}$. Then D gives $\texttt{share}_i$ to P_i as a share for s.

Reconstruction algorithm: A set of parties $A \in \Sigma$ can reconstruct s if and only if $(1, 0, \cdots, 0)$ is in the linear span of

$$\mathcal{M}_A = \{\mathbf{V}_j \mid \psi(j) \in A\},$$

where $\mathbf{V}_j$ denotes the jth row of $\mathcal{M}$. If this is indeed the case then there exists a *recombination vector* $\boldsymbol{\alpha}_A$, such that $\boldsymbol{\alpha}_A \cdot \mathcal{M}_A = (1, 0, \ldots, 0)$. Let $\boldsymbol{s}_A$ denote the set of shares corresponding to the parties in A. Then these parties can reconstruct s by computing $s = \langle \boldsymbol{\alpha}_A, \boldsymbol{s}_A^T \rangle$, where $\langle x, y \rangle$ denotes the *dot product* of x and y.

Definition 2 (Monotone Span Programme (MSP) [3]). *We say that the above $(\mathcal{M}, \psi)$ is a MSP which realizes Σ and the size of the MSP is ℓ.*

Theorem 1 ([3]). *The above algorithm constitutes a valid secret sharing scheme.*

Theorem 2 ([3]). *Two different secrets shared according to an MSP realizing Σ cannot have same shares corresponding to an access set.*

Notice that there may be more than one row of $\mathcal{M}$ assigned to a party P_i. However, as assumed in [3], for the ease of presentation, we assume that each P_i is assigned exactly one row in $\mathcal{M}$, namely $\mathbf{V}_i$. This is without loss of generality. Finally we use the following notation throughout our paper.

Notation 1 *Let $\mathcal{R}$ be any subset of $\mathcal{P}$ i.e $\mathcal{R} \subseteq \mathcal{P}$. Then $\mathcal{M}_{\mathcal{R}}$ denotes the matrix containing the rows of $\mathcal{M}$ corresponding to the parties in $\mathcal{R}$.*

2.3 Verifiable Secret Sharing (VSS)

In the definition of secret sharing (see Sec. 2.1), we assumed that $D \notin \mathcal{A}$ and the parties under the control of $\mathcal{A}$ *honestly* follows the protocol. However these are very restricting assumptions. A VSS scheme relaxes these assumptions. In a VSS protocol, $D \in \mathcal{P}$, holds a secret $s \in \mathbb{F}$. The protocol consists of a sharing phase and a reconstruction phase. During the protocol, a *computationally unbounded* adversary $\mathcal{A}$ can select any set $B \in \Gamma$ (possibly including D) for corruption. Moreover, the corrupted parties can behave in *any arbitrary manner*. Now we call the protocol a VSS protocol if it satisfies the following conditions:

1. **Secrecy**: If D is *honest*, then s will be information theoretically secure during the sharing phase.
2. **Correctness**: If D is *honest*, then the honest parties will output s at the end of the reconstruction phase, irrespective of the behavior of $\mathcal{A}$.
3. **Strong Commitment**: If D is *corrupted*, then at the end of the sharing phase there is a value $s^\star \in \mathbb{F}$, such that at the end of the reconstruction phase all honest parties will output $s^\star$, irrespective of the behavior of $\mathcal{A}$.

3 Two Round VSS Tolerating $\mathcal{Q}^4$ Adversary Structure

Let $\mathcal{A}$ be a non-threshold adversary, characterized by an adversary structure Γ, such that $\mathcal{A}$ satisfies the $\mathcal{Q}^4$ condition. Moreover let $\mathcal{M}$ be the $n \times e$ MSP realizing the corresponding access structure $\Sigma = \Gamma^c$. We then present a two round VSS protocol tolerating $\mathcal{A}$. Before presenting our protocol, we give the following definition:

Definition 3 ($\mathcal{A}$-clique). *Let $G = (V, E)$ be an undirected graph, where $V = \mathcal{P}$ and let C be a clique in G. Moreover, let V_C denote the vertices belonging to C. Then we say that C is an $\mathcal{A}$-clique in G if $V \setminus V_C \in \Gamma$. That is, the set $B = V \setminus V_C$ belongs to the adversary structure.*

Algorithm for Finding $\mathcal{A}$-clique: Let $\Gamma = \{B_1, \ldots, B_{|\Gamma|}\}$. For $i = 1, \ldots, \Gamma$, check whether the parties in $\mathcal{P} \setminus B_i$ form a clique in G, which requires *polynomial computation*. If yes, then the algorithm terminates and $\mathcal{P} \setminus B_i$ is the $\mathcal{A}$-clique. If there exists no $B_i \in \Gamma$ such that $\mathcal{P} \setminus B_i$ form a clique in G then there is no $\mathcal{A}$-clique. *This algorithm performs computation, which is polynomial in $|\Gamma|$.*

Our two round protocol is given in Fig. 1.

We now proceed to prove the properties of the protocol. In the proof, we will use the following notations:

- Let ShHo (resp. ShB) denote the set of honest (resp. corrupted) parties in Sh at the end of sharing phase when sharing phase is successful.
- Let ReHo (resp. ReB) denote the set of honest (resp. corrupted) parties in Rec.

Lemma 1. *An honest D will never be discarded during sharing phase.*

Fig. 1. Two Round VSS for Sharing a Secret s Tolerating $\mathcal{A}$

Sharing Phase

Round I:

1. D selects a random, symmetric $e \times e$ matrix R, such that $R[1,1] = s$.
2. D computes $u_i = \mathbf{V}_i \cdot R$ and sends u_i to P_i. The first entry of u_i, denoted by s_i, is referred as i^{th} share of s, given to P_i. Moreover, $\langle u_i, \mathbf{V}_j \rangle$, for $j = 1, \ldots, n$, is referred as j^{th} share-share of s_i, denoted by s_{ij}.
3. For $i = 1, \ldots, n-1$, party P_i selects a random r_{ij} for every P_j, where $j > i$ and privately sends r_{ij} to P_j.

Round II:

1. For $i = 1, \ldots, n$, party P_i broadcasts the following, for each $j \neq i$:
   - $a_{ij} = r_{ij} + \langle u_i, \mathbf{V}_j \rangle = r_{ij} + s_{ij}$, if $j > i$;
   - $a_{ij} = r_{ji} + \langle u_i, \mathbf{V}_j \rangle = r_{ji} + s_{ij}$, if $j < i$;

Local Computation (By Each Party):

1. Construct an undirected graph G_{Sh} over $\mathcal{P}$, where there exists an edge (P_i, P_j), for $j > i$, if $a_{ij} = a_{ji}$. Notice that all honest parties will construct the *same* G_{Sh}.
2. If no $\mathcal{A}$-clique is present in G_{Sh} then the *sharing phase fails* and D is *discarded*. [a]
3. If there is an $\mathcal{A}$-clique in G_{Sh}, then *sharing phase succeeds*. Let Sh denote the parties in $\mathcal{A}$-clique and let $\mathsf{Sh\text{-}Del} = \mathcal{P} \setminus \mathsf{Sh}$. Notice that all honest parties will find the *same* $\mathcal{A}$-clique and hence the same Sh.

Reconstruction Phase

Round I:

1. Each party $P_i \in \mathsf{Sh}$ broadcasts u_i received from D. Let it be denoted by $\overline{u}_i$.

Local Computation (By Each Party):

1. Construct an undirected graph G_{Rec} over the set of parties in Sh, where there exists an edge (P_i, P_j), for $j > i$, if both $P_i, P_j \in \mathsf{Sh}$ and $\langle \overline{u}_i, \mathbf{V}_j \rangle = \langle \overline{u}_j, \mathbf{V}_i \rangle$.
2. Find $\mathcal{A}$-clique (which is bound to exist) in G_{Rec}. Let Rec denote the parties in $\mathcal{A}$-clique and let $\mathsf{Rec\text{-}Del} = \mathsf{Sh} \setminus \mathsf{Rec}$. Notice that all honest parties will find the *same* $\mathcal{A}$-clique and hence the set Rec.
3. Without loss of generality, let $P_1, \ldots, P_{|\mathsf{Rec}|}$ be the parties in Rec and let $\overline{s}_1, \ldots, \overline{s}_{|\mathsf{Rec}|}$ be the shares (the first entry of $\overline{u}_i$'s) revealed by these parties. Then reconstruct $\overline{s}$ by applying reconstruction algorithm of the LSSS to $\overline{s}_1, \ldots, \overline{s}_{|\mathsf{Rec}|}$ and terminate.

[a] Following the convention of [5, 4, 7], if D is discarded during the sharing phase, then some pre-defined value from $\mathbb{F}$ is taken as D's secret.

PROOF: If D is honest, then $\langle u_i, \mathbf{V}_j \rangle = \langle u_j, \mathbf{V}_i \rangle$ and hence $a_{ij} = a_{ji}$ will hold for each honest P_i, P_j. So the set of honest parties will form an $\mathcal{A}$-clique in G_{Sh} and D will not be discarded. □

Lemma 2. *If the sharing phase succeeds, then* ShHo *is an access set. Moreover,* $\langle u_i, \mathbf{V}_j \rangle = \langle u_j, \mathbf{V}_i \rangle$ *will hold for each* $P_i, P_j \in \mathsf{ShHo}$*, where* $i < j$.

Proof: It is easy to see that $\mathsf{ShHo} \cup \mathsf{ShB} \cup \mathsf{Sh\text{-}Del} = \mathcal{P}$. If the sharing phase succeeds, then $\mathsf{Sh\text{-}Del} \in \Gamma$. Also $\mathsf{ShB} \in \Gamma$. Now if $\mathsf{ShHo} \in \Gamma$, then it implies that $\mathcal{A}$ does not satisfy $\mathcal{Q}^3$ (and hence $\mathcal{Q}^4$) condition, which is a contradiction. The second part of the lemma follows from the fact if $P_i, P_j \in \mathsf{ShHo}$, then $a_{ij} = a_{ji}$ and both P_i and P_j would have honestly used r_{ij}. □

Lemma 3. *Without loss of generality, let* $\mathsf{ShHo} = \{P_1, \ldots, P_t\}$*. If the sharing phase succeeds, then there exists a vector* $\boldsymbol{x} = (s^\star, \boldsymbol{\rho})$*, for some* $\boldsymbol{\rho} \in \mathbb{F}^{e-1}$*, such that*

$$(s_1, \ldots, s_t)^T = \mathcal{M}_{ShHo} \cdot \boldsymbol{x}^T.$$

In other words, the shares of the parties in ShHo *will be valid shares of* $s^\star$*, such that* D *will be committed to* $s^\star$*. Moreover, if* D *is honest then* $s^\star = s$.

Proof: From the previous lemma, if the sharing phase succeeds, then for each $P_i, P_j \in \mathsf{ShHo}$, we have $s_{ij} = s_{ji}$. Let $S_{ShHo} = \{s_{ij}\}$ be the $t \times t$ symmetric matrix. Then S_{ShHo} can be expressed as

$$S_{ShHo} = \mathcal{M}_{ShHo} \cdot U_{ShHo} = U_{ShHo}^T \cdot \mathcal{M}_{HaHo}^T,$$

where $U_{ShHo} = [\boldsymbol{u_1}^T, \ldots, \boldsymbol{u_t}^T]$. Also from the previous lemma, ShHo is an access set. Therefore, there exists a recombination vector $\boldsymbol{\alpha}_{ShHo}$, such that $\boldsymbol{\alpha}_{ShHo} \cdot \mathcal{M}_{ShHo} = (1, 0, \ldots, 0)$. Hence,

$$\boldsymbol{\alpha}_{ShHo} \cdot S_{ShHo} = \boldsymbol{\alpha}_{ShHo} \cdot \mathcal{M}_{ShHo} \cdot U_{ShHo} = (1, 0, \ldots, 0) \cdot U_{ShHo} = (s_1, \ldots, s_t).$$

On the other hand,

$$\boldsymbol{\alpha}_{ShHo} \cdot S_{ShHo} = \boldsymbol{\alpha}_{ShHo} \cdot U_{ShHo}^T \cdot \mathcal{M}_{ShHo}^T = \boldsymbol{x} \cdot \mathcal{M}_{ShHo}^T,$$

where $\boldsymbol{x} = \boldsymbol{\alpha}_{ShHo} \cdot U_{ShHo}^T$. Therefore, $(s_1, \ldots, s_t) = \boldsymbol{x} \cdot \mathcal{M}_{ShHo}^T = \mathcal{M}_{ShHo} \cdot \boldsymbol{x}^T$.

If D is honest then $s^\star = s$. Because, in this case, $\boldsymbol{x} = \boldsymbol{\alpha}_{ShHo} \cdot U_{ShHo}^T = \boldsymbol{\alpha}_{ShHo} \cdot \mathcal{M}_{ShHo} \cdot R = (1, 0, \ldots, 0) \cdot R$, which is nothing but the first row of R.□

Lemma 4. *If the sharing phase succeeds, then an* $\mathcal{A}$*-clique will be present in* G_{Rec}.

Proof: From Lemma 2, ShHo is an access set and for each $P_i, P_j \in \mathsf{ShHo}$, we have $\langle u_i, \mathbf{V}_j \rangle = \langle u_j, \mathbf{V}_i \rangle$. During the reconstruction phase, each $P_i, P_j \in \mathsf{ShHo}$ will correctly broadcast $\overline{u}_i = u_i$ and $\overline{u}_j = u_j$ respectively. So during the reconstruction phase also, $\langle u_i, \mathbf{V}_j \rangle = \langle u_j, \mathbf{V}_i \rangle$ will hold. Thus ShHo will always form an $\mathcal{A}$-clique in G_{Rec}. □

Lemma 5. *If the sharing phase succeeds, then* ReHo *will be an access set. Moreover, the shares of the parties in* ReHo *will define the same secret* $s^\star$*, as committed by* D *to the parties in* ShHo *during the sharing phase.*

Proof: Notice that $\mathsf{ReHo} \cup \mathsf{ReB} \cup \mathsf{Rec\text{-}Del} \cup \mathsf{Sh\text{-}Del} = \mathcal{P}$. Now we know that $\mathsf{Sh\text{-}Del}, \mathsf{Rec\text{-}Del} \in \Gamma$. Also $\mathsf{ReB} \in \Gamma$. Now if $\mathsf{ReHo} \in \Gamma$, then it implies that $\mathcal{A}$ does not satisfy $\mathcal{Q}^4$ condition, which is a contradiction. The second part of the lemma follows from the fact that $\mathsf{ReHo} \subseteq \mathsf{ShHo}$. □

Lemma 6. *During the reconstruction phase, every $P_i \in \mathsf{Rec}$ will correctly disclose s_i, the i^{th} share of the secret $s^\star$, which is committed by D during the sharing phase to the parties in ShHo.*

Proof: The lemma holds trivially when $P_i \in \mathsf{Rec}$ is *honest*. We now consider the case when $P_i \in \mathsf{Rec}$ is *corrupted*. Before proceeding further, notice that P_i will have an edge with each of the parties in ReHo in graph G_{Rec}, since the set of parties in Rec forms a clique. This further implies that $\overline{u}_i$ disclosed by P_i satisfies $\langle \overline{u}_i, \mathbf{V}_j \rangle = \langle \overline{u}_j, \mathbf{V}_i \rangle$, for each $P_j \in \mathsf{ReHo}$. That is, $s_{ij} = s_{ji}$, for each $P_j \in \mathsf{ReHo}$. Also $\overline{u}_j = u_j$, for each $P_j \in \mathsf{ReHo}$. For simplicity assume that ShHo and ReHo contains the first t and y parties respectively, where $y \leq t$. Now from Lemma 3, we know that there exists $\boldsymbol{x} = (s^\star, \boldsymbol{\rho})$, such that

$$(s_1, \ldots, s_t)^T = \mathcal{M}_{ShHo} \cdot \boldsymbol{x}^T$$

Now following the notations as used in Lemma 3, we also have

$$(s_1, \ldots, s_y)^T = \mathcal{M}_{ReHo} \cdot \boldsymbol{x}^T$$

Now $(s_1, \ldots, s_y)^T = \mathcal{M}_{ReHo} \cdot \boldsymbol{x}^T$ implies that $\boldsymbol{x} \cdot \mathcal{M}_{ReHo}^T = \boldsymbol{\alpha}_{ReHo} \cdot U_{ReHo}^T \cdot \mathcal{M}_{ReHo}^T$. This is because $(s_1, \ldots, s_y)^T = \mathcal{M}_{ReHo} \cdot \boldsymbol{x}^T$ implies that

$$\begin{aligned}
\boldsymbol{x} \cdot \mathcal{M}_{ReHo}^T &= (s_1, \ldots, s_y) \quad \text{(taking transpose on both sides)} \\
&= (1, 0, \ldots, 0) \cdot U_{ReHo} \\
&= \boldsymbol{\alpha}_{ReHo} \cdot \mathcal{M}_{ReHo} \cdot U_{ReHo} \\
&= \boldsymbol{\alpha}_{ReHo} \cdot U_{ReHo}^T \cdot \mathcal{M}_{ReHo}^T
\end{aligned}$$

Here $\boldsymbol{\alpha}_{ReHo}$ is the recombination vector corresponding to the access set ReHo and $U_{ReHo} = [\boldsymbol{u}_1^T, \ldots, \boldsymbol{u}_y^T]$. Now we will show that $s_i = \overline{u}_{i1}$, as revealed by corrupted $P_i \in \mathsf{Rec}$ is the i^{th} share of $s^\star$. That is, $s_i = \boldsymbol{x} \cdot \mathbf{V}_i^T = \mathbf{V}_i \cdot \boldsymbol{x}^T$. Now notice that, $\boldsymbol{\alpha}_{ReHo} \cdot \mathcal{M}_{ReHo} = (1, 0, \ldots, 0)$. It is easy to see that

$$\boldsymbol{\alpha}_{ReHo} \cdot [s_{i1}, \ldots, s_{iy}]^T = \overline{u}_{i1} \tag{3}$$

Now we will show that following also is true:

$$\boldsymbol{\alpha}_{ReHo} \cdot [s_{1i}, \ldots, s_{yi}]^T = \boldsymbol{x} \cdot \mathbf{V}_i^T \tag{4}$$

We start with the known equation:

$$S_{ReHo} = U_{ReHo}^T \cdot \mathcal{M}_{ReHo}^T$$

Here $S_{ReHo} = \{s_{ij} : 1 \leq i, j \leq y\}$ is the symmetric matrix. Now pre-multiplying both the sides of above equation by $\boldsymbol{\alpha}_{ReHo}$, we get

$$\boldsymbol{\alpha}_{ReHo} \cdot S_{ReHo} = \boldsymbol{\alpha}_{ReHo} \cdot U_{ReHo}^T \cdot \mathcal{M}_{ReHo}^T$$

Now we know that $\boldsymbol{\alpha}_{ReHo} \cdot U_{ReHo}^T \cdot \mathcal{M}_{ReHo}^T = \boldsymbol{x} \cdot \mathcal{M}_{ReHo}^T$. So substituting in the above equation, we get

$$\boldsymbol{\alpha}_{ReHo} \cdot S_{ReHo} = \boldsymbol{x} \cdot \mathcal{M}_{ReHo}^T$$

Both the sides of the above equation turns out to be some row vector of equal length. Now concentrating on the value of the i^{th} index of the row vectors in the above equation, we get $\boldsymbol{\alpha}_{ReHo} \cdot [s_{1i}, \ldots, s_{yi}]^T = \boldsymbol{x} \cdot \mathbf{V}_i^T$. Now as discussed above, $s_{ij} = s_{ji}$, for $j = 1, \ldots, y$. So left hand side of Eqn. 3 and Eqn. 4 are same. Thus s_i revealed by $P_i \in \mathsf{Rec}$ is the i^{th} share of $s^\star$. □

Now using the above lemmas, we prove the following theorem.

Theorem 3. *The protocol in Fig. 1 is a two round VSS scheme tolerating $\mathcal{A}$, satisfying the $\mathcal{Q}^4$ condition. The communication cost is polynomial in the size of $\mathcal{M}$, and the computation cost is polynomial in the size of Γ.*

Proof: We only show that the protocol satisfies all the properties of VSS, as round, computation and communication complexity are easy to verify.

1. **Secrecy**: We have to only consider the case when D is honest. Let the adversary corrupt some $B \in \Gamma$. Then at the end of **Round I** of the sharing phase, adversary learns no information about s from their shares, as B is a non-access set. Let $i \notin B$ and $j \notin B$. Then at the end of **Round I** of the sharing phase, the adversary gains no information about r_{ij}. Hence at the end of **Round II**, adversary gains no information about u_i, as r_{ij} or r_{ji} works as the one-time pad. Thus, at the end of the sharing phase, s remains information theoretically secure (see [3] for complete details).
2. **Correctness**: We have to consider the case when D is honest. If D is honest then the sharing phase will succeed. Now the parties in ShHo is an access set and defines s. Moreover, correct share of s will be revealed by every P_i in Rec. These facts guarantee that by applying the reconstruction algorithm of the LSSS to the shares of the parties in Rec, s will be reconstructed correctly.
3. **Strong Commitment**: We have to consider the case when D is corrupted. The proof is very similar to the proof of correctness. In this case, the parties in ShHo is an access set and defines some secret $s^\star$, which is D's committed secret. Moreover, ReHo is an access set where $\mathsf{ReHo} \subseteq \mathsf{ShHo}$ and hence define the same secret $s^\star$. Furthermore, correct share of $s^\star$ will be revealed by every P_i in Rec. These facts guarantee that by applying the reconstruction algorithm of the LSSS to the shares of the parties in Rec, secret $s^\star$ will be reconstructed correctly and uniquely. □

4 Three Round VSS Tolerating $\mathcal{Q}^3$ Adversary Structure

We first design a three round *weak commitment scheme* (WCS) protocol.

4.1 Three Round WCS Tolerating $\mathcal{Q}^3$ Adversary Structure

In a WCS, there exists a dealer $D \in \mathcal{P}$, who has a secret $s \in \mathbb{F}$, which he wants to commit to the parties in $\mathcal{P}$. The scheme consists of two phases as follows:

1. **Commit phase**: Initially, D has a secret s. At the end of the commit phase, either D is discarded (by all honest parties) or s is committed.
2. **Decommit phase**: If D is not discarded during the commit phase then:
   – D broadcasts (s, ρ), where ρ is the randomness used by D during the commit phase.
   – Each P_i broadcasts its view w_i of the commit phase.
   – Then a validity check function Valid is applied which outputs either *valid* or *invalid*.

   We say that s is accepted as *authentic* if

$$\mathsf{Valid}(s, \rho, w_1, \cdots, w_n) = \text{ valid.}$$

A protocol is a WCS scheme tolerating $\mathcal{A}$ if the following conditions are satisfied:

1. **Secrecy**: If D is *honest*, then $\mathcal{A}$ obtains no information about s during the commit phase.
2. **Correctness**: If D is *honest* then s will be accepted as authentic during the decommit phase.
3. **Weak Commitment**: If D is *corrupted* and not discarded during the commit phase, then there exists an $s^\star \in \mathbb{F}$, such that D is committed to $s^\star$ during the commit phase. Moreover, if some s' is accepted as authentic during the decommit phase, then $s' = s^\star$.

The *round complexity* of a WCS scheme is the number of communication rounds during the commit phase. We now present our three round WCS in Fig. 2.
We now show that the scheme presented in Fig. 2 is a valid WCS scheme, tolerating $\mathcal{A}$, provided $\mathcal{A}$ satisfies the $\mathcal{Q}^3$ condition. In the proofs, we use the following notations:

– Let HaHo (resp. HaB) denote the set of happy and honest (resp. happy and corrupted) parties at the end of commit phase if commit phase is successful.
– Let WCoHo (resp. WCoB) denote the set of honest (resp. corrupted) parties in $WCORE$ if decommit phase is successful.

Lemma 7. *If D is honest, then D will not be discarded during the commit phase. Moreover, s will be accepted as authentic during the decommit phase.*

Proof: By easy inspection we note that the set $\mathsf{UnHappy}$ contains only corrupted parties, when D is honest. Thus $\mathsf{UnHappy} \in \Gamma$ and so the commit phase succeeds.

Now to show that s will be accepted as authentic during the decommit phase, we prove that $\mathcal{P} \setminus WCORE \in \Gamma$ during the decommit phase. To begin with, an honest D will correctly broadcast $\boldsymbol{x'} = \boldsymbol{x}$ and each honest P_i will correctly broadcast $s'_i = s_i$. Thus, all honest parties will be present in $WCORE$ and hence $\mathcal{P} \setminus WCORE$ will contain only corrupted parties. Hence $\mathcal{P} \setminus WCORE \in \Gamma$. Thus the decommit phase will also succeed and s will be accepted as authentic. □

Fig. 2. Three Round WCS for Committing a Secret s

Commit Phase

Round I:

1. D selects a random, symmetric $e \times e$ matrix R, such that $R[1,1] = s$. Let $\boldsymbol{x} = (s, \boldsymbol{\rho})$ be the first column (and row) of R.
2. D computes $u_i = \mathbf{V}_i \cdot R$ and privately sends u_i to party P_i. The first entry of u_i, denoted by s_i, is referred as the share of s, given to party P_i. Moreover, $\langle u_i, \mathbf{V}_j \rangle$ is referred as the j^{th} share-share of s_i, denoted by s_{ij}.
3. Party P_i, for $i = 1, \ldots, n-1$, selects a random pad r_{ij}, for each $j > i$ and privately sends r_{ij} to party P_j.

Round II:

1. For $i = 1, \ldots, n$, party P_i broadcasts the following, for each $j \neq i$:
   - $a_{ij} = r_{ij} + \langle u_i, \mathbf{V}_j \rangle = r_{ij} + s_{ij}$, if $j > i$;
   - $a_{ij} = r_{ji} + \langle u_i, \mathbf{V}_j, \rangle = r_{ji} + s_{ij}$, if $j < i$;

Round III:

1. For each pair (i, j), such that $j > i$, if $a_{ij} \neq a_{ji}$, then
   - P_i broadcasts $\alpha_{ij} = \langle u_i, \mathbf{V}_j \rangle$;
   - P_j broadcasts $\beta_{ji} = \langle u_j, \mathbf{V}_i \rangle$;
   - D broadcasts $\gamma_{ij} = \langle u_i, \mathbf{V}_j \rangle = \langle u_j, \mathbf{V}_i \rangle$.

   Party P_i (P_j) is said to be *unhappy*, if the value broadcasted by him, mismatches the value broadcasted by D.

Local Computation (By Each Party):

1. Let UnHappy be the set of unhappy parties. If UnHappy $\in \Gamma$, then the commit phase succeeds. Otherwise, *commit phase fails* and D is discarded.

Decommit Phase

Round I:

1. D broadcasts the first row of R used by him during the sharing phase. Let it be denoted by $\boldsymbol{x'}$ and let s' be the first entry of $\boldsymbol{x'}$.
2. Each *happy* party P_i broadcasts the share received by him from D during the sharing phase. Let it be denoted by s'_i.

Local Computation (By Each Party):

1. Let $WCORE$ be the set of all such *happy* P_i's, such that $\boldsymbol{x'} \cdot \mathbf{V}_i^T = s'_i$. In other words, a happy $P_i \in WCORE$ if s'_i is a valid share of s' according to the LSSS.
2. If $\mathcal{P} \setminus WCORE \in \Gamma$, then *decommit succeeds* and so accept s' as authentic.
3. If $\mathcal{P} \setminus WCORE \notin \Gamma$, then *decommit fails* and so do not accept s' as authentic.

Lemma 8. *If the commit phase succeeds, then* HaHo *is an access set. Moreover, for each* $P_i, P_j \in$ HaHo*, where* $i < j$, $\langle u_i, \mathbf{V}_j \rangle = \langle u_j, \mathbf{V}_i \rangle$.

Proof: It is easy to see that $\mathsf{HaHo} \cup \mathsf{HaB} \cup \mathsf{UnHappy} = \mathcal{P}$. If the commit phase succeeds, then $\mathsf{UnHappy} \in \Gamma$. Also $\mathsf{HaB} \in \Gamma$. This implies that $\mathsf{HaHo} \notin \Gamma$, otherwise $\mathcal{A}$ does not satisfy $\mathcal{Q}^3$ condition, which is a contradiction. The second part follows from easy inspection. □

Lemma 9. *Without loss of generality, let $\mathsf{HaHo} = \{P_1, \ldots, P_t\}$. If the commit phase succeeds, then there exists a vector $\boldsymbol{x}^\star = (s^\star, \rho)$, where $\rho \in \mathbb{F}^{e-1}$ such that*

$$(s_1, \ldots, s_t)^T = \mathcal{M}_{HaHo} \cdot \boldsymbol{x}^{\star T}.$$

In other words, D will commit the secret $s^\star$ to the parties in HaHo. Moreover, if D is honest then $\boldsymbol{x}^\star = \boldsymbol{x}$, where $\boldsymbol{x}$ is the first column of R used by D during the sharing phase and hence $s^\star = s$.

Proof: Follows using similar arguments as used in Lemma 3. □

Lemma 10. *If the decommit phase succeeds, then WCoHo is an access set and $\langle u_i, \mathbf{V}_j \rangle = \langle u_j, \mathbf{V}_i \rangle$ for each $P_i, P_j \in \mathsf{WCoHo}$. Furthermore, the shares of the parties in WCoHo define the same secret as defined by shares of the parties in HaHo.*

Proof: Notice that $\mathsf{WCoHo} \cup \mathsf{WCoB} \cup (\mathcal{P} \setminus WCORE) = \mathcal{P}$. If the decommit phase succeeds, then $\mathcal{P} \setminus WCORE \in \Gamma$. Also, $\mathsf{WCoB} \in \Gamma$. This implies that $\mathsf{WCoHo} \notin \Gamma$, otherwise $\mathcal{A}$ does not satisfy the $\mathcal{Q}^3$ condition. The second and third part follows from Lemma 8 and the fact that $\mathsf{WCoHo} \subseteq \mathsf{HaHo}$. □

Theorem 4. *The protocol in Fig. 2 is a three round WCS scheme where the honest parties perform computation and communication, polynomial in $\mathcal{M}$.*

Proof: We only show that the protocol satisfies the properties of WCS scheme. The other properties follows easily from inspection.

1. **Secrecy**: Follows using similar arguments as used in our two round VSS.
2. **Correctness**: Follows from Lemma 7.
3. **Weak Commitment**: We have to consider the case when D is *corrupted*. If decommit phase fails, then it satisfies weak commitment. On the other hand, if decommit succeeds and s' is accepted as authentic then it implies that for each $P_i \in WCORE$, $\boldsymbol{x'} \cdot \mathbf{V}_i^T = s'_i = \mathbf{V}_i \cdot \boldsymbol{x'}^T$, where $\boldsymbol{x'} = [s', \rho']$. This will also be true for each party in WCoHo. Without loss of generality, assume that the first y parties are present in WCoHo. The parties in WCoHo are honest implies $s_i = s'_i$ for $i = 1, \ldots, y$. Therefore we have $(s_1, \ldots, s_y)^T = \mathcal{M}_{WCoHo} \cdot \boldsymbol{x'}^T$. Also from Lemma 9, we have $(s_1, \ldots, s_y)^T = \mathcal{M}_{WCoHo} \cdot \boldsymbol{x}^{\star T}$, where $\boldsymbol{x}^\star = [s^\star, \rho]$. Now this imply that $s' = s^\star$ because WCoHo is an access set and two different secrets cannot have same shares corresponding to an access set (see Theorem 2). Hence, the accepted secret s' is the same secret $s^\star$, as committed by D to $\mathsf{WCoHo} \subseteq \mathsf{HaHo}$. □

Fig. 3. Three Round VSS for Sharing Secret s Tolerating $\mathcal{A}$

Sharing Phase

Round I:

1. D performs the first two steps as in the commit phase of three round WCS.
2. Each party P_i selects a random value r^i and starts executing an instance of three round WCS protocol to commit r^i, as a dealer. We denote the i^{th} instance of WCS as WCS_i. Let $r_1^i, \ldots, r_n^i$ denote the shares of r^i generated in WCS_i, such that P_i has given r_j^i to P_j during **Round I** of WCS_i.

Round II:

1. For $i = 1, \ldots, n$, party P_i broadcasts the following, for each $j \neq i$: $a_{ij} = r_j^i + \langle u_i, \mathbf{V}_j \rangle = r_j^i + s_{ij}$; and $b_{ij} = r_i^j + \langle u_i, \mathbf{V}_j \rangle = r_i^j + s_{ij}$.
2. Concurrently, **Round II** of WCS_i is executed, for $i = 1, \ldots, n$.

Round III:

1. For each pair (i, j), such that $a_{ij} \neq b_{ji}$, parties do the following:
   - P_i broadcasts $\alpha_{ij} = \langle u_i, \mathbf{V}_j \rangle$; P_j broadcasts $\beta_{ji} = \langle u_j, \mathbf{V}_i \rangle$ and D broadcasts $\gamma_{ij} = \langle u_i, \mathbf{V}_j \rangle = \langle u_j, \mathbf{V}_i \rangle$.

   P_i (P_j) is *unhappy*, if α_{ij} (β_{ji}) mismatches γ_{ij}.
2. Concurrently, **Round III** of WCS_i is executed, for $i = 1, \ldots, n$.

Local Computation (By Each Party):

1. Let Sh be the set of *happy* parties such that their instance of the commit phase of WCS as a dealer is successful. Let Ha_i denote the set of happy parties in the sharing phase of WCS_i for $P_i \in \mathsf{Sh}$.
2. Continue to keep a party P_i in Sh if $\mathcal{P} \setminus (\mathsf{Sh} \cap \mathsf{Ha}_i) \in \Gamma$. Otherwise remove P_i from Sh.
3. Repeat the previous step, till no more parties can be removed from Sh. Now if $\mathcal{P} \setminus \mathsf{Sh} \in \Gamma$, then *the sharing phase succeeds*. Otherwise, it fails and D is discarded.

Reconstruction Phase

Round I:

1. For each $P_i \in \mathsf{Sh}$, run the decommit phase of WCS_i.
2. Every $P_i \in \mathsf{Sh}$ broadcasts the vector obtained from D. Let it be denoted by $\overline{u_i}$.

Local Computation (By Each Party):

1. Let Rec be the set of parties P_i from Sh, such that both the following hold:
   - The decommit phase of WCS_i is successful, with output say $\overline{r^i}$ being accepted as authentic. Let $WCORE_i$ be the $WCORE$, corresponding to WCS_i and let $\overline{r_j^i}$ be the share of $\overline{r^i}$, as disclosed by $P_j \in WCORE_i$ during the decommit phase of WCS_i.
   - Compute $\overline{s_{ij}}$ for every $P_j \in WCORE_i$ as follows:
     (a) $\overline{s_{ij}} = \gamma_{ij}$; if γ_{ij} was broadcasted by D during **Round III** of the sharing phase.
     (b) $\overline{s_{ij}} = a_{ij} - \overline{r_j^i}$; if γ_{ij} was not broadcasted by D during **Round III** of the sharing phase. Here a_{ij} was broadcasted by P_i during the sharing phase.

     Now the computed $\overline{s_{ij}}$'s corresponding to each $P_j \in WCORE_i$ must be consistent with $\overline{u_i}$. Precisely $\overline{s_{ij}} = \langle \overline{u_i}, \mathbf{V}_j \rangle$ must hold, for every $P_j \in WCORE_i$.
2. For every $P_i \in \mathsf{Rec}$, assign $\overline{s_i} = \overline{u_{i1}}$, where $\overline{u_{i1}}$ is the first entry of $\overline{u_i}$.
3. Compute $\overline{s}$ from $\overline{s_i}$'s corresponding to $P_i \in \mathsf{Rec}$ using reconstruction algorithm of LSSS.

4.2 Three Round VSS Tolerating $\mathcal{Q}^3$ Adversary Structure

Now we design our three round VSS (given in Fig. 3) using our three round WCS as a black-box. We now prove the properties of the VSS protocol. For the proof, we use the following notations:

- Let ShHo (resp. ShB) denote the set of honest (resp. corrupted) parties in Sh at the end of sharing phase when the sharing phase is successful.
- ReHo (resp. ReB) denote the set of honest (resp. corrupted) parties in Rec.

Lemma 11. *If D is honest then the sharing phase will always succeed.*

PROOF: To show that the sharing phase succeeds for an honest D, we prove that $\mathcal{P} \setminus \mathsf{Sh} \in \Gamma$. This is proved by showing that an honest party can never be in $\mathcal{P} \setminus \mathsf{Sh}$ and therefore $\mathcal{P} \setminus \mathsf{Sh}$ contains only a set of corrupted parties. First we note that each honest party P_i will be happy and their instance of WCS will be successful and Ha_i will include all honest parties. Naturally, $\mathcal{P} \setminus (\mathsf{Sh} \cap \mathsf{Ha}_i)$ contains only corrupted parties and will belong to Γ. Thus, all honest parties will be present in Sh. Equivalently, $\mathcal{P} \setminus \mathsf{Sh}$ contains only a set of corrupted parties. □

Lemma 12. *If the sharing phase succeeds, then ShHo is an access set. Moreover, for each $P_i, P_j \in \mathsf{ShHo}$, $\langle u_i, \mathbf{V}_j \rangle = \langle u_j, \mathbf{V}_i \rangle$. Furthermore, without loss of generality, let $\mathsf{ShHo} = \{P_1, \ldots, P_t\}$. Then there exists a vector $\boldsymbol{x} = (s^\star, \rho)$, such that*

$$(s_1, \ldots, s_t)^T = \mathcal{M}_{ShHo}.\boldsymbol{x}^T.$$

In other words, D will commit the secret $s^\star$ to the parties in ShHo during the sharing phase. Moreover, if D is honest then $s^\star = s$.

PROOF: Follows using similar arguments as used in our two round VSS and three round WCS. □

Lemma 13. *If the sharing phase succeeds then $\mathsf{ShHo} = \mathsf{ReHo}$.*

PROOF: During the reconstruction phase, every honest $P_i \in \mathsf{Sh}$ will correctly broadcast the vector which it received from D during sharing phase. So we have $\overline{u_i} = u_i$. Now from the correctness property of WCS scheme, the decommit phase of WCS_i, corresponding to the honest P_i will be successful and r^i will be accepted as authentic. So we have $\overline{r^i} = r^i$ and also $\overline{r^i_j} = r^i_j$ for every $P_j \in WCORE_i$. Hence the computed $\overline{s_{ij}}$ will be equal to $s_{ij} = \langle u_i, \mathbf{V}_j \rangle$. So the honest $P_i \in \mathsf{Sh}$ will be present in Rec. Therefore the lemma holds. □

Lemma 14. *For every $P_i \in \mathsf{Rec}$, $\overline{s_i}$ computed during reconstruction phase, is same as the i^{th} share of secret $s^\star$, which is defined by the shares of the parties in ShHo (and hence ReHo).*

PROOF: From the previous two lemmas, the shares of the parties in $\mathsf{ShHo} = \mathsf{ReHo}$ will define a unique secret $s^\star$, which is D's committed secret. Now we have the following two cases:

1. $P_i \in \mathsf{Rec}$ is *honest*: In this case, the lemma holds trivially.

2. $P_i \in \mathsf{Rec}$ is *corrupted*: Since $P_i \in \mathsf{Rec}$, it implies that decommit phase of WCS_i is successful and hence r^i which was committed by P_i during commit phase is accepted as authentic. Now $P_i \in \mathsf{Rec}$ also implies that $\mathcal{P} \setminus (\mathsf{Sh} \cap \mathsf{Ha}_i) \in \Gamma$. Now let CoH_i be the set of *common honest* parties in $(\mathsf{Sh} \cap \mathsf{Ha}_i)$. It is easy to see that CoH_i is an access set, otherwise $\mathcal{A}$ will not satisfy $\mathcal{Q}^3$ condition, which is a contradiction. Now $\mathsf{CoH}_i \subseteq \mathsf{ShHo} = \mathsf{ReHo}$. Also, $\mathsf{CoH}_i \subseteq \mathsf{WCORE}_i \subseteq \mathsf{Ha}_i$. Thus, r^i_j revealed by every $P_j \in \mathsf{CoH}_i$ during decommit phase of WCS_i is the correct share of r^i, as given by P_i to P_j during commit phase of WCS_i. Thus, the computed $\overline{s_{ij}}$, corresponding to every $P_j \in \mathsf{CoH}_i$ is equal to s_{ji}. This is because there can be either one of the following two possibilities:
   (a) Both P_i and P_j are happy during sharing phase, but $a_{ij} \neq b_{ji}$. In this case, $\overline{s_{ij}} = \gamma_{ij} = \beta_{ji} = s_{ji}$;
   (b) Both P_i and P_j are happy during sharing phase and $a_{ij} = b_{ji}$. In this case, $\overline{s_{ij}} = a_{ij} - r^i_j = b_{ji} - r^i_j = s_{ji}$

   Now the shares of the parties in CoH_i define the same secret $s^\star$. This is because, as discussed above, the access set $\mathsf{CoH}_i \subseteq \mathsf{ReHo}$. Since CoH_i is an access set, from the properties of MSP, it follows that s_{ji}'s corresponding to $P_j's \in \mathsf{CoH}_i$ uniquely define s_i, the i^{th} share of the committed secret $s^\star$ (this can be shown using the same arguments as used in Lemma 6).

   On the other hand, $P_i \in \mathsf{Rec}$ also implies that $\overline{u_i}$ revealed by P_i is consistent with all $\overline{s_{ij}} = s_{ji}$'s of $P_j \in \mathsf{CoH}_i$. This further implies that $\overline{u_{i1}}$ is same as s_i because CoH_i is an access set (again this can be shown using the same arguments as used in Lemma 6). □

Theorem 5. *The protocol in Fig. 3 is a three round VSS tolerating non-threshold adversary $\mathcal{A}$ characterized by an adversary structure Γ, where $\mathcal{A}$ satisfies the $\mathcal{Q}^3$ condition. In the protocol, the honest parties perform computation and communication which is polynomial in the size of $\mathcal{M}$.*

PROOF: The round complexity can be verified by inspection. Also, it is easy to see that the honest parties perform computation and communication which is polynomial in the size of $\mathcal{M}$. We now show that the protocol satisfies the properties of VSS.

1. **Secrecy**: We have to only consider the case when D is honest. Let the adversary corrupt some $B \in \Gamma$. Then at the end of **Round I** of the sharing phase, adversary learns no information about s from their shares, as B is a non-access set. From the secrecy property of WCS, the adversary will not get any information about r^i's, which are committed by honest P_i's. Hence, at the end of **Round I** of sharing phase, the adversary gains no information about r^i_j's and r^j_i's, corresponding to $P_i, P_j \notin B$. Hence at the end of **Round II**, adversary gains no information about u_i and u_j, as r^i_j's and r^j_i's works as the one-time pad.

   During **Round III**, if $a_{ij} \neq b_{ji}$ or vice-versa, then P_i or P_j is corrupted (as D is honest). Hence, the adversary already knows the share-share $\langle u_i, \mathbf{V}_j \rangle =$

$\langle u_j, \mathbf{V}_i \rangle$. Thus, D's broadcast of γ_{ij} during **Round III** adds no extra information about u_i to adversary's view. Thus, at the end of sharing phase, s remains information theoretically secure.

2. **Correctness**: We have to consider the case when D is honest. If D is honest then the sharing phase will succeed (see Lemma 11). Now by Lemma 12, the parties in ShHo is an access set and defines s. Moreover, by Lemma 14, correct share of s will be reconstructed for every P_i in Rec. These facts guarantee that by applying reconstruction algorithm of the LSSS to the shares of the parties in Rec, secret s will be reconstructed correctly.
3. **Strong Commitment**: We have to consider the case when D is corrupted. The proof is very similar to the proof of correctness. By Lemma 12, the parties in ShHo is an access set and defines some secret $s^\star$, which is D's committed secret. Moreover, from Lemma 13, $\mathsf{ShHo} = \mathsf{RecHo}$. Furthermore, by Lemma 14, correct share of s^* will be reconstructed for every P_i in Rec. These facts guarantee that by applying reconstruction algorithm of the LSSS to the shares of the parties in Rec, secret $s^\star$ will be reconstructed correctly and uniquely. □

5 Lower Bounds

We now give our lower bound results.

Theorem 6. *Two round perfectly secure VSS is possible if and only if $\mathcal{A}$ satisfies the $\mathcal{Q}^4$ condition.*

Proof: Sufficiency follows from Fig. 1. We now prove the necessity. On the contrary, assume that a two round VSS protocol, say Π, is possible even though $\mathcal{A}$ does not satisfy the $\mathcal{Q}^4$ condition. This implies that there exists B_1, B_2, B_3 and B_4, belonging to the underlying adversary structure Γ, such that $B_1 \cup B_2 \cup B_3 \cup B_4 = \mathcal{P}$. Now consider protocol Π', involving parties P_1, P_2, P_3 and P_4, where party P_i performs the same computation and communication, as done by the parties in B_i in Π, for $i = 1, \ldots, 4$. It is easy to see that if Π is a two round VSS protocol, then $\Pi' s$ is also a two round VSS protocol involving four parties, out of which at most one can be corrupted. However, from [5], Π' does not exist. So Π also does not exist. □

Theorem 7. *Any r-round perfectly secure VSS protocol, where $r \geq 3$, is possible if and only if $\mathcal{A}$ satisfies the $\mathcal{Q}^3$ condition.*

Proof: Follows using similar arguments as used in Theorem 6 and by the result of [5]. □

6 Flaw in the Reconstruction Phase of VSS of [4]

In [4], the authors presented a three round VSS tolerating a threshold adversary $\mathcal{A}_t$ with $n = 3t+1$, using a three round WSS protocol as a black-box. However, we now show that there is a flaw in the reconstruction phase of their VSS. Moreover, we also show the modifications to eliminate this flaw. We start with a brief discussion on the WSS and VSS of [4]. *Here we use slightly different notations and steps, that were not there in [4]. However, the current discussion will be valid even with the original notations and steps of [4].* The sharing phase of the WSS of [4] is a special case of the commit phase of our WCS. Precisely the matrix $\mathcal{M}$ here is an $n \times (t+1)$ *Vandermonde* matrix, whose i^{th} row is $[i^0, i^1, \ldots, i^t]$ and R is the coefficient matrix of a random symmetric bi-variate polynomial $F(x, y)$ of degree-t in x, y, where $F(0, 0) = s$. The result of the computation in the WSS of [4] can be viewed as follows (though this view was not presented in [4], the essence is same): if D is not discarded during sharing phase, then there exists a degree-t univariate polynomial, say $f(x)$, such that D has WSS-shared $f(x)$ and each *happy and honest* party P_i has received $f(i)$ from D. Moreover, if D is honest then D will not be discarded and $f(x) = f_0(x) = F(x, 0)$ and hence $f(0) = s$. Now during reconstruction phase, either $f(x)$ (and hence $f(0) = s$) or $NULL$ will be reconstructed. Moreover, if $f(x)$ is reconstructed then it is reconstructed with the shares revealed by a set of parties WCORE, such that WCORE is a subset of *happy* parties and there exists at least $t+1$ honest parties in WCORE.

Now the VSS protocol of [4] works as follows: During the sharing phase, D selects a random symmetric bi-variate polynomial $F(x, y)$ of degree-t in x, y, where $F(0, 0) = s$ and gives each P_i, the degree-t polynomial $f_i(x) = F(x, i)$. Then the parties perform *pair-wise* checking to check the consistency of their common values. To do this, each party P_i acts as a dealer and WSS-shares a degree-t polynomial $f_i^W(x)$ and gives each P_j the share $f_i^W(j)$. Now to do the consistency checking, each P_i broadcasts $a_{ij} = f_i(j) + f_i^W(j)$ and $b_{ij} = f_i(j) + f_j^W(i)$. Each inconsistency (i.e., $a_{ij} \neq b_{ji}$) is resolved by D (by broadcasting $f_i(j)$), as a result of which parties become *happy*/*unhappy* and the computation proceeds. At the end of sharing phase, all honest parties agree on a set of at least $2t+1$ *happy* parties, say $CORE_{Sh}$, such that the following condition holds:

1. For each $P_i, P_j \in CORE_{Sh}$, we have $f_i(j) = f_j(i)$;
2. Each $P_i \in CORE_{Sh}$ as a dealer, has AWSS-shared a degree-t polynomial $f_i^W(x)$ to at least $2t+1$ parties in $CORE_{Sh}$.

Now notice that there is a subtle point here, which is the basis of the flaw in the reconstruction phase of VSS protocol of [4]. *Even though $f_i(j) = f_j(i)$ is true for every $P_i, P_j \in CORE_{Sh}$ (as both of them are happy), it does not imply that $a_{ij} = b_{ji}$ is true for every $P_i, P_j \in CORE_{Sh}$*. Obviously, if both $P_i, P_j \in CORE_{Sh}$ are *honest*, then $a_{ij} = b_{ji}$. However, if at least one of $P_i, P_j \in CORE_{Sh}$ is *corrupted*, then it may happen that $a_{ij} \neq b_{ji}$, but still both P_i and P_j are happy and are present in $CORE_{Sh}$. More concretely, suppose P_i is *corrupted*, P_j and D are *honest*. Then during **Round II** of sharing phase, P_i may broadcast a_{ij} that is not equal to b_{ji}. But during **Round III**, when D tries to resolve the inconsistency, P_i may broadcast correct $f_i(j)$. That is D broadcasts $\gamma_{ij} = f_i(j)$, P_i broadcasts $\alpha_{ij} = f_i(j)$ and P_j broadcasts $\beta_{ji} = f_j(i)$, such that $\gamma_{ij} = \alpha_{ij} = \beta_{ji}$. So both P_i and P_j will be *happy*. Moreover P_i as a dealer can behave correctly during his instance of WSS to share $f_i^W(x)$, such that P_i satisfies the second property stated above to be in $CORE_{Sh}$.

We now recall the steps of the reconstruction phase of the VSS protocol of [4] in Fig. 4. In [4], the authors claimed that reconstructed $f_i(x)$'s of any $t+1$ parties

Fig. 4. Reconstruction Phase of the VSS Protocol of [4]

For each $P_i \in CORE_{Sh}$, run the reconstruction phase of WSS_i (the instance of WSS initiated by P_i as a dealer).

Local Computation (By Each Party):

1. Initialize $CORE_{Rec} = CORE_{Sh}$.
2. Remove P_i from $CORE_{Rec}$ if the reconstruction phase of WSS_i outputs $NULL$.
3. If $f_i^W(x)$ is reconstructed during reconstruction phase of WSS_i then compute $f_i(j) = a_{ij} - f_i^W(j)$, for $j = 1, \ldots, n$. Check if the computed $f_i(j)$'s lie on a unique degree-t polynomial. If not then remove P_i from $CORE_{Rec}$. Otherwise, let $f_i(x)$ be the degree-t polynomial.
4. Take $f_i(x)$'s corresponding to any $t+1$ parties in $CORE_{Rec}$, reconstruct $F^\star(x, y)$ and output $s^\star = F^\star(0, 0)$.

in $CORE_{Rec}$ define the same bivariate polynomial of degree-t in x and y (see Lemma 6 of [4]). However, we now show that this is not the case. To be precise, consider a setting where D is *honest* and P_i is *corrupted*. During **Round I** of sharing phase, P_i gets $f_i(x) = F(x, i)$. Then P_i as a dealer WSS-shares a degree-t polynomial $f_i^W(x)$. During **Round II**, P_i broadcasts $a_{ij} = f_i'(j) + f_i^W(j)$, instead of $f_i(j) + f_i^W(j)$, corresponding to all P_j's, such that $f_i'(x) \neq f_i(x)$ is another degree-t polynomial. So $a_{ij} \neq b_{ji}$, for *all* P_j's. But then during **Round III**, P_i behaves in such a way that P_i is considered as *happy* along with all other P_j's (this he can do as discussed earlier). P_i also ensures that his WSS instance satisfies the desired property so that P_i is included in $CORE_{Sh}$.

Now during reconstruction phase of VSS, suppose the reconstruction phase of WSS_i is successful and hence the WSS-shared polynomial $f_i^W(x)$ is reconstructed correctly. But now when the (honest) parties perform step 3 of the local computation (given in Fig. 4), they will get back $f_i'(j) = a_{ij} - f_i^W(j)$, instead of original $f_i(j)$. Moreover, the computed $f_i'(j)$'s will lie on degree-t polynomial $f_i'(x) \neq f_i(x)$ and P_i will be present in $CORE_{Rec}$. But now notice that $f_i'(x) \neq f_i(x)$ does not lie on the original bivariate polynomial $F(x, y)$. This will further lead to the violation of correctness property of VSS.

Elimination of the Flaw: From the above discussion, it is clear that the reason behind the above flaw is that $a_{ij} = b_{ji}$ may not hold for every $P_i, P_j \in CORE_{Sh}$. To eliminate the above flaw, we modify the step 3 of the local computation of Fig. 4 as follows:

3. If $f_i^W(x)$ is reconstructed during reconstruction phase of WSS_i then compute $f_i(j)$'s as follows:

- $f_i(j) = \gamma_{ij}$; if γ_{ij} was broadcasted by D during **Round III** of sharing phase.
- $f_i(j) = a_{ij} - f_i^W(j)$; if $a_{ij} = b_{ji}$ during sharing phase.

Check if the computed $f_i(j)$'s lie on a unique degree-t polynomial. If not then remove P_i from $CORE_{Rec}$. Otherwise, let $f_i(x)$ be the degree-t polynomial.

Now it is easy to verify that with the above modification, Lemma 6 of [4] will hold.

7 More Efficient 3-round VSS for $n \geq 3t + 1$

In the previous section, we pointed out a flaw in the 3-round VSS of Fitzi et al. [4], and presented how to fix it. The communication complexity of the reconstruction phase of the proposed modified protocol is $\mathcal{O}(n^3)$. This results from the facts that there are n instances of the WSS protocol in the VSS and the communication cost of the reconstruction phase of WSS of [4] is $\mathcal{O}(n^2)$.

On the other hand, if we restrict our three round VSS protocol given in Fig. 3 to threshold adversary, then we get a three round VSS with $n = 3t + 1$ whose communication complexity of reconstruction phase is $\mathcal{O}(n^2)$. This results from the facts that in our VSS, WSS has been replaced by WCS and the communication cost of the decommit phase of WCS is only $\mathcal{O}(n)$. If we compare the definition of WCS and WSS (for formal definition of WSS, see [4]), then we find that in WSS, the dealer D is not allowed to act/play a special role in the reconstruction phase. That is, D is not allowed to reveal the secret and randomness used by him during the sharing phase. During the reconstruction phase, every party reveal their entire view of the sharing phase and a reconstruction function is applied on them to reconstruct either the secret shared during sharing phase or $NULL$. On the other hand, in WCS, D is allowed to act specially in the decommit phase. Precisely, he is allowed to reveal the secret and randomness used by him during commit phase. As a result, the decommit phase of our WCS is conceptually simpler than the reconstruction phase of WSS protocol of [4] and we gain an efficiency of $\Theta(n)$ during the reconstruction phase.

8 Conclusion

In this paper, we resolved the round complexity of VSS tolerating generalized adversary. Our results strictly generalize the results of [4] to non-threshold settings. In our three round protocol, we have not tried to optimize the use of broadcast channel. However, we conjecture that following the techniques of [7], we can design a three round VSS tolerating $\mathcal{Q}^3$ adversary structure, which uses broadcast channel in only one round during the sharing phase.

Acknowledgement: We would like to thank the anonymous referees of ICITS 2011 for several useful comments.

References

1. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation (Extended Abstract). In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 1–10. ACM Press, 1988.
2. B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults (Extended Abstract). In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 383–395. ACM Press, 1985.
3. R. Cramer, I. Damgård, and U. M. Maurer. General Secure Multi-party Computation from any Linear Secret Sharing Scheme. In B. Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 316–334. Springer Verlag, 2000.
4. M. Fitzi, J. Garay, S. Gollakota, C. Pandu Rangan, and K. Srinathan. Round-Optimal and Efficient Verifiable Secret Sharing. In S. Halevi and T. Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 329–342. Springer Verlag, 2006.
5. R. Gennaro, Y. Ishai, E. Kushilevitz, and T. Rabin. The Round Complexity of Verifiable Secret Sharing and Secure Multicast. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece. ACM*, pages 580–589. ACM Press, 2001.
6. M. Hirt and U. M. Maurer. Complete Characterization of Adversaries Tolerable in Secure Multi-Party Computation. In *Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing, Santa Barbara, California, USA, August 21-24, 1997*, pages 25–34. ACM Press, 1997.
7. J. Katz, C. Koo, and R. Kumaresan. Improving the Round Complexity of VSS in Point-to-Point Networks. In L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, volume 5126 of *Lecture Notes in Computer Science*, pages 499–510. Springer Verlag, 2008.
8. F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error Correcting Codes.* North-Holland Publishing Company, 1978.
9. U. M. Maurer. Secure multi-party computation made simple. In S. Cimato, C. Galdi, and G. Persiano, editors, *Security in Communication Networks, Third International Conference, SCN 2002, Amalfi, Italy, September 11-13, 2002. Revised Papers*, volume 2576 of *Lecture Notes in Computer Science*, pages 14–28. Springer, 2002.
10. A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.