Does Pseudo-basis Extend to General Adversary?

Ashish Choudhury[*1] **Kaoru Kurosawa**[2]
Arpita Patra[3]

[1]Applied Statistics Unit
Indian Statistical Institute Kolkata India
partho31@gmail.com, partho_31@yahoo.co.in

[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

Pseudo-basis is a powerful and fundamental concept in coding theory, introduced by Kurosawa and Suzuki (EUROCRYPT '08), which allows to efficiently localize t errors in a codeword in an interactive fashion, even by using a linear error correcting code with distance *only* $t+1$. It is used to construct the first efficient, communication and round optimal, 2-round *perfectly secure message transmission* (PSMT) scheme for $n = 2t+1$, where an infinitely powerful adversary can *actively* corrupt t out of n (bidirectional) channels between a sender $\mathbf{S}$ and a receiver $\mathbf{R}$. Recently, Yang and Desmedt (ASIACRYPT '10) claimed that by using pseudo-basis, they constructed the first ever *efficient, constant round* PSMTs against general (non-threshold) adversaries in undirected as well as directed networks.

In this paper, our main contribution is to show that pseudo-basis *can not* be extended to general adversaries. This means that the PSMTs of Yang and Desmedt (based on pseudo-basis) are flawed. We next show a *truly efficient, three round* PSMT over directed networks against general adversaries without using pseudobasis. While the previous PSMT schemes in directed network designed without using pseudobasis are either *inefficient* or prone to *guessing attack* (shown by Yang and Desmedt in ICITS '09), our efficient scheme can resist guessing attack. Instead of using pseudo-basis, we generalize the *union technique* to our setting, which was previously used to construct PSMTs against threshold proactive adversaries over undirected networks. We also present simple and efficient 3-round scheme in undirected network, without basing it on pseudobasis, which can send multiple messages concurrently.

Keywords: Pseudo-basis, general adversary, Information Theoretic Security.

1 Introduction

In the model of information-theoretically secure message transmission schemes, there exists a *computationally unbounded malicious* adversary who can corrupt some subset of N channels (also called *wires*) between a sender $\mathbf{S}$ and a receiver $\mathbf{R}$. $\mathbf{S}$ wishes to send a message $m^{\mathbf{S}}$ to $\mathbf{R}$ without sharing any key in such a way that *perfect secrecy* and *perfect reliability* are satisfied. Such schemes are called *perfectly secure message transmission* (PSMT) schemes and were first incepted by Dolev et al. [7]. Since then it has been studied extensively in various network model and adversarial settings [21, 5, 23, 1, 18, 8, 13, 4, 2] due to its fundamental importance in secure distributed computing.

[*]Financial support from Department of Information Technology, Government of India acknowledged

Suppose that the adversary can corrupt at most t out of N channels (t-active *threshold adversary*) in an undirected network (where all the N channels are bidirectional). Then for $N \geq 3t+1$, 1-round PSMT can be easily constructed by using Shamir's (t, N)-threshold secret sharing scheme [22]. First *perfect secrecy* is satisfied from the definition of (t, N)-threshold secret sharing schemes. Second *perfect reliability* is satisfied because Shamir's scheme is equivalent to Reed-Solomon code with the minimum Hamming distance [14]

$$d = N - t \geq 3t + 1 - t = 2t + 1.$$

Hence $\mathbf{R}$ can correct t errors caused by the adversary. However, this simple technique cannot be used for $N \leq 3t$ any more because $d < 2t + 1$ in this case.

Pseudo-basis and It's Applications. Now for any linear code with the minimum Hamming distance $d \geq t + 1$, *pseudo-basis* [13] suggests a brilliant way to locate all error locations in two rounds of interaction. That is if $\mathbf{R}$ sends a sequence of codewords to $\mathbf{S}$ in round one, and $\mathbf{S}$ replies back to $\mathbf{R}$ with the pseudo-basis of the received vectors [1] in round two, then $\mathbf{R}$ can locate every error that happened in round one. Further the pseudo-basis computation requires only polynomial time.

Essentially, pseudo-basis exploits the fact that *errors in the received vectors are not random* i.e. when a set of codewords are sent together, the error locations in the vectors at the receiving end are not at all random, rather they concentrate over the t locations corrupted by t-bounded adversary. Therefore, the error vectors (the difference of received vectors and the corresponding codewords) are bound to be spanned over a space of dimension t. This important observation is at the heart of pseudo-basis. Pseudo-basis also proved to be extremely communication efficient when employed in PSMT.

The notion of pseudo-basis was introduced by Kurosawa and Suzuki [13] to construct the first ever *efficient*, simultaneously *communication and round optimal*, two round PSMT tolerating a t-active *threshold adversary*. Thus apart from being a fundamental contribution to the coding theory, pseudo-basis turns out to be an extremely powerful tool for simultaneously reducing round and communication complexity of PSMT protocols. The 2-round protocol of [13] is the culmination of the works of Srinathan et al. [23] who presented a 2-round PSMT protocol with simultaneously optimal round and communication complexity ($\mathcal{O}(n\ell)$ field elements for a message of ℓ field elements [23]) and subsequent work of Agarwal et al. [1] who showed the protocol of [23] to be flawed and then presented a protocol for the same purpose, but at the expense of exponential communication cost, where the message size has to be impractically super-polynomial.

On the other hand, pseudobasis is used to repair secrecy leak in the existing PSMT protocols in directed networks. In a directed network, $\mathbf{S}$ is connected to $\mathbf{R}$ by some n unidirectional wires (called as *top band*) and $\mathbf{R}$ is connected to $\mathbf{S}$ by u unidirectional wires (called as *bottom band*). All existing efficient PSMT schemes over directed network [5, 15, 20] were vulnerable to *guessing attack* [24], that breaks the perfect secrecy of protocols. Informally, adversary mounts a guessing attack by making random guesses for some secrets, sending them to $\mathbf{S}$ over the bottom band and then watching the reaction of $\mathbf{S}$ on his guesses, which is broadcasted over top band in the next round. Essentially, adversary manages to get an oracle service from $\mathbf{S}$ to know whether his guesses about some secret are correct or not. The guessing attacks rely on the fact that there may not be sufficient connectivity in the bottom band for reliable communication to be possible. Pseudo-basis is proved to offer a way to get rid of *guessing attack* and design efficient protocols in directed networks against t-active adversary [17].

General Adversary. Since it is a very useful technique, it is worthwhile to ask whether the concept of pseudo-basis can be generalized for general errors specified by general adversary [9, 10, 6, 3], that most often captures the adversarial behavior in more realistic way. A *general adversary* (also called non-threshold adversary), denoted by $\mathcal{A}$ is characterized by an *adversary structure* Γ, which is a collection of subsets of channels $\mathcal{W}$ that the adversary $\mathcal{A}$ can *potentially* corrupt. That is, $\Gamma = \{B \subset \mathcal{W} \mid \mathcal{A} \text{ can corrupt } B\}$. A t-active threshold adversary is a *special case* of general adversary, such that $|B| \leq t$, for each $B \in \Gamma$.

In a novel attempt, Yang and Desmedt [25] tried to address the above issue. They claimed the answer to be affirmative and used the generalized pseudo-basis for constructing efficient, constant round PSMT protocols in undirected and directed networks tolerating general adversary, which are first of their kind.

Our Contribution. In this paper, we make multiple contributions on PSMT against general adversaries.

[1] At the receiving end, the codewords turn to vectors of length N due to introduction of unwanted errors.

1. We show that the concept of pseudo-basis cannot be generalized to general adversary settings, in contrary to what is claimed in [25]. Specifically, we prove that pseudo-basis does not locate the identities of all corrupted channels, when used against general adversary. This allows us to show that the efficient 2-round and 3-round PSMTs for undirected network and 3-round PSMTs for directed network in [25] which use pseudo-basis are no longer correct. This means there is now no *efficient* 3-round PSMT in directed network for general adversary. Our next contribution fills this gap.
2. We present the first *efficient, constant round* PSMT for directed networks against general adversaries tolerating guessing attack, which satisfies **N&S-II** condition (see **Related work for the meaning of N&S-II condition**). Our protocol runs in 3 rounds, and does not base on pseudo-basis. Interestingly, *union technique* introduced in [19] for the construction of PSMT tolerating threshold proactive adversary in undirected networks, shows to have the power to foil guessing attack. We extend union technique for general adversary and employ it for our protocol design.
3. We also present a new and simple 3-round efficient PSMT in undirected networks against general adversaries, satisfying $\mathcal{Q}^2$ condition, without using pseudo-basis (see **Related work for the meaning of $\mathcal{Q}^2$ condition**). Our protocol can send multiple field elements concurrently and it has the same communication complexity as that of 3-round protocol of [12], which can send *only* one field element.

Our contributions are summarized in the following table.

	Undirected network	Directed network
Threshold adversary	pseudo-basis [13]	pseudo-basis [17]
General adversary	We show [25] is flawed and present new PSMT for $\mathcal{Q}^2$	We show [25] is flawed and present new PSMT for **N&S-II**

Related Works on PSMT for General Adversary. It is harder to design efficient PSMT protocols for directed networks than for undirected networks. We now discuss the existing results for PSMT against general adversary in both undirected and directed networks. We first give the following definition:

Definition 1.1 ($\mathcal{Q}^k$ Condition [9])**.** We say that an adversary $\mathcal{A}$ characterized by an adversary strcuture Γ, satisfies $\mathcal{Q}^k$ condition with respect to a set of channels $\mathcal{W}$, if there exists no k sets in Γ, which adds upto the whole set $\mathcal{W}$. That is: $\forall B_1, \ldots, B_k \in \Gamma : B_1 \cup \ldots \cup B_k \neq \mathcal{W}$.

Undirected networks: Here all the N wires between **S** and **R** are *bidirectional*.

[6] showed that 1-round PSMT is possible if and only if the adversary $\mathcal{A}$ satisfies $\mathcal{Q}^3$ condition. Recently [12] presented an efficient single round PSMT protocol tolerating $\mathcal{Q}^3$ adversary structure. If *multiple rounds* are allowed, then PSMT protocol is possible if and only if $\mathcal{A}$ satisfies $\mathcal{Q}^2$ condition [10]. There are several *multi round* protocols in undirected networks tolerating $\mathcal{Q}^2$ adversary structure [10, 12, 25]. The protocol of [10] shows the possibility of PSMT in $\mathcal{O}(n)$ rounds tolerating $\mathcal{Q}^2$ adversary. With minimum number of rounds i.e. two, [12] presents an inefficient protocol, while [25] presents an efficient protocol. [25] also presents an efficient 2-round protocol for sending multiple messages *concurrently*, which is flawed, as it uses pseudo-basis. In this paper, we present an efficient 3-round protocol for multiple messages without using pseudo-basis. Though [25] also presents an efficient 3-round protocol for multiple messages, it is incorrect because of the use of pseudo-basis.

Directed networks: Here there are total $N = n + u$ wires between **S** and **R** among which n channels, denoted by $\mathcal{Y} = \{w_1, \ldots, w_n\}$ are directed from **S** to **R**, also called as *top band* and u channels, denoted by $\mathcal{X} = \{q_1, \ldots, q_u\}$ are directed from **R** to **S**, also called as *bottom band*. Thus $\mathcal{W} = \mathcal{Y} \cup \mathcal{X}$.

In *directed network*, the necessary and sufficient condition for one, two and more than two rounds protocol are different. A 1-round PSMT in directed network is essentially the same as a 1-round PSMT in undirected network. Therefore $\mathcal{Q}^3$ condition with respect to $\mathcal{Y}$ is necessary and sufficient. Following are the conditions for the other two cases:

Condition for 2-round PSMT: For threshold adversaries, 2-round PSMT in directed networks is possible if and only if $n + u \geq 3t + 1$ and $n \geq 2t + 1$ [16].

For general adversaries, it is possible if and only if $\mathcal{A}$ satisfies $\mathcal{Q}^2$ condition with respect to $\mathcal{Y}$, and $\mathcal{A}$ satisfies $\mathcal{Q}^3$ condition with respect to $\mathcal{Y} \cup \mathcal{X} = \mathcal{W}$ [25]. We call this condition as **N&S-I**..

Condition for r-round PSMT, where $r \geq 3$: For threshold adversaries, r-round PSMT in directed networks, where $r \geq 3$, is possible if and only if $n + 2u \geq 3t + 1$ and $n \geq 2t + 1$ [5].

For general adversaries, it is possible if and only if the following holds [20]: (i) $\mathcal{A}$ satisfies $\mathcal{Q}^2$ condition with respect to $\mathcal{Y}$ and (ii) if any $A_1, A_2, A_3 \in \Gamma$ satisfies $A_1 \cup A_2 \cup A_3 \supset \mathcal{Y}$, then at most one of these three sets may cover the entire $\mathcal{X}$. That is, there should not exist A_i, A_j, where $i \neq j$ and $i, j \in \{1, 2, 3\}$, such that $\mathcal{X} \subseteq A_i$ and $\mathcal{X} \subseteq A_j$. We call this condition as **N&S-II**..

Intuitively, the size of bottom band $\mathcal{X}$ is smaller in **N&S-II** than in **N&S-I**. Hence it is harder to design PSMT protocols for **N&S-II** than for **N&S-I**. For **N&S-I**, [25] presents an efficient 2-round protocol without using pseudobasis. However for **N&S-II**, the only known 3-round protocol without using pseudobasis has *exponential* (in size of Γ) communication and round complexity [24]. The other existing multi (more than 2) round protocols are either vulnerable to guessing attack ([20]) or incorrect due to the use of pseudo-basis (the 3-round protocols of [25]). Therefore there is no efficient 3-round protocol in directed networks. We settle this in our paper.

Organization. In Section 2, we recall the concept of pseudo-basis from [13]. Section 3 shows that pseudo-basis fails to achieve its properties against general adversary. This section also contains discussion on monotone span program (MSP), Linear Codes, Linear Secret Sharing Scheme (LSSS) and their equivalence and their relation with general adversary. In Section 4, we propose a new, efficient 3-round PSMT scheme tolerating general adversary in *directed network* satisfying **N&S-II**, while Section 5 informally describes 3-round PSMT for multiple messages in *undirected network*. We conclude with some interesting open problems in Section 6.

2 Pseudo-basis in Threshold Settings

We now recall the notion of pseudo-basis from [13]. The setting is as follows: **A** and **B** are connected by N bi-directional channels, denoted by $\mathcal{W} = \{w_1, \ldots, w_N\}$. Here **A** can be **S** (**R**) and accordingly, **B** will be **R** (**S**). Moreover, $N \geq 2t+1$, which is a necessary and sufficient condition for the existence of any r-round PSMT protocol tolerating t-active threshold adversary $\mathcal{A}_t$, where $r \geq 2$ [7]. For the current discussion, we fix $N = 2t + 1$. Let $\mathcal{C}$ a linear code over $\mathbb{F}$ of length N, such that minimum Hamming distance [14] of $\mathcal{C}$ is $t + 1$. In [13], the authors have considered $\mathcal{C}$ to be Reed-Solomon code [14] to help the intuition. Thus $\mathcal{C}$ is the set of all possible Reed-Solomon (RS) codewords of length N over $\mathbb{F}$, which are encoded using polynomials of degree t over $\mathbb{F}$. Moreover, let $\alpha_1, \ldots, \alpha_N$ be the publicly known elements from $\mathbb{F}$, at which the polynomials are evaluated to compute the RS codewords in $\mathcal{C}$. Thus $\mathcal{C}$ is the set of all possible vectors $(p(\alpha_1), \ldots, p(\alpha_N))$, where $p(x)$ is a polynomial over $\mathbb{F}$ having degree at most t. Now suppose **A** sends several codewords, say γ codewords $C_1, \ldots, C_\gamma \in \mathcal{C}$ to **B** over the N channels by transmitting i^{th} component of all the codewords over i^{th} channel w_i. During the transmission, $\mathcal{A}_t$ can control t channels and can change the component(s) of the codeword(s) sent over those channels. The locations at which error(s) occur in the codewords are not completely random, since they occur in the codewords within some fixed set of t locations. The pseudo-basis exploits this fact.

Let **B** receive the set of vectors $\mathcal{Z} = \{Y_1 \ldots, Y_\gamma\}$ over the N channels, such that $Y_i = C_i + E_i$, for $i = 1, \ldots, \gamma$. Here $E_i = (e_{i1}, \ldots, e_{iN})$ is an error vector introduced by the adversary. Since at most t errors could occur, each E_i have at most t non-zero components. Let $support(E_i) = \{j \mid e_{ij} \neq 0\}$. Then there is a set of t wires $\{w_{i_1}, \ldots, w_{i_t}\} \subset \mathcal{W}$ that are controlled by $\mathcal{A}_t$, such that each error vector E_i satisfies $support(E_i) \subseteq \{i_1, \ldots, i_t\}$. This means that the space E spanned by the error vectors $E_1, \ldots, E_\gamma$ has dimension at most t. The notion of pseudo-basis exploits this idea. For two vectors $V_1, V_2 \in \mathbb{F}^N$, we write

$$V_1 = V_2 \ mod \ \mathcal{C} \quad \text{if} \quad V_1 - V_2 \in \mathcal{C}.$$

Notice that for $i = 1, \ldots, \gamma$, for every triplet (Y_i, C_i, E_i), $Y_i = E_i \bmod \mathcal{C}$.

Definition 2.1 (Linearly Pseudo-Express [13])**.** We say that a vector $Y \in \mathcal{Z}$ is linearly pseudo-expressed by $\{B_1, \ldots, B_k\}$ if there exists some $\alpha = (a_1, \ldots, a_k)$, such that $Y = a_1B_1 + \ldots + a_kB_k \bmod \mathcal{C}$.

The following lemma is very crucial, since we will show that if part of the lemma falls apart for general adversary settings. *It is due to this reason, pseudo-basis fails to achieve its properties in general adversary settings.* We recall the proof for if part from [13].

Lemma 2.2 ([13])**.** *Let $\mathcal{E} = \{E_1, \ldots, E_\gamma\}$ be the error vector set of $\mathcal{Z}$. Then $E_i \in \mathcal{E}$ is linearly expressed by $\{E_{j_1}, \ldots, E_{j_k}\} \subseteq \mathcal{E}$ if and only if Y_i is linearly pseudo-expressed by $\{Y_{j_1}, \ldots, Y_{j_k}\}$.*

Proof. (if part.) Suppose that Y_i is linearly pseudo-expressed by $\{Y_{j_1}, \ldots, Y_{j_k}\}$. Then we have $Y_i = (a_1 Y_{j_1} + \ldots + a_k Y_{j_k}) \; mod \; \mathcal{C}$ for some $a_1, \ldots, a_k$. This implies that:

$$\begin{aligned} (\mathcal{C}_i + E_i) &= \{a_1(\mathcal{C}_{j_1} + E_{j_1}) + \ldots + a_k(\mathcal{C}_{j_k} + E_{j_k})\} \; mod \; \mathcal{C} \\ &\Rightarrow (\mathcal{C}_i - a_1\mathcal{C}_{j_1} - \ldots - a_k\mathcal{C}_{j_k}) + (E_i - a_1 E_{j_1} - \ldots - a_k E_{j_k}) \in \mathcal{C} \end{aligned}$$

By linearity property of $\mathcal{C}$, we have $(\mathcal{C}_i - a_1\mathcal{C}_{j_1} - \ldots - a_k\mathcal{C}_{j_k}) \in \mathcal{C}$ and $(E_i - a_1 E_{j_1} - \ldots - a_k E_{j_k}) \in \mathcal{C}$. As the number of non-zero components in each $E_i, E_{j_1}, \ldots, E_{j_k}$ is at most t, the vector $(E_i - a_1 E_{j_1} - \ldots - a_k E_{j_k})$ will have at least $N - t \geq t+1$ zero components. However, in $\mathcal{C}$, there is only one codeword, namely 0-codeword (i.e. a tuple of length N containing all 0's) $(0, \ldots, 0)$, which has at least $t+1$ zero components. This is because distance of $\mathcal{C}$ is at least $t+1$ and hence every non-zero codeword of $\mathcal{C}$ must have at least $t+1$ non-zero components. This imply that $(E_i - a_1 E_{j_1} - \ldots - a_k E_{j_k})$ is an all zero vector (0-codeword) which further implies that $E_i = a_1 E_{j_1} + \ldots + a_k E_{j_k}$. This means that if Y_i is linearly pseudo-expressed by $\{Y_{j_1}, \ldots, Y_{j_k}\}$, then E_i is linearly expressed by $\{E_{j_1}, \ldots, E_{j_k}\}$. $\square$

We now complete the rest of the theory of pseudo-basis.

Definition 2.3 (Pseudo-Span, Pseudo-Basis and Pseudo-Dimension [13])**.** We say that $\{Y_{j_1} \ldots, Y_{j_k}\} \subset \mathcal{Z}$ pseudo-spans $\mathcal{Z}$ if each $Y_i \in \mathcal{Z}$ can be linearly pseudo-expressed by $\{Y_{j_1} \ldots, Y_{j_k}\}$. We say that $\{Y_{j_1} \ldots, Y_{j_k}\} \subset \mathcal{Z}$ is a pseudo-basis of $\mathcal{Z}$ if it is the minimum subset of $\mathcal{Z}$ which pseudo-spans $\mathcal{Z}$. If $\{Y_{j_1} \ldots, Y_{j_k}\} \subset \mathcal{Z}$ is a pseudo-basis of $\mathcal{Z}$ and $k = |\{Y_{j_1} \ldots, Y_{j_k}\}|$, then we say that $\mathcal{Z}$ has pseudo-dimension k.

Theorem 2.4 ([13])**.** *Let $\mathcal{E} = \{E_1, \ldots, E_\gamma\}$ be the error vector set of $\mathcal{Z}$. Then $\mathcal{B}_e = \{E_{j_1}, \ldots, E_{j_k}\} \subset \{E_1, \ldots, E_\gamma\}$ is a basis of $\mathcal{E}$ if and only if $\mathcal{B}_y = \{Y_{j_1}, \ldots, Y_{j_k}\} \subset \mathcal{Z}$ is a pseudo-basis of $\mathcal{Z}$ (Note that $\mathcal{B}_e$ and $\mathcal{B}_y$ have same indices). In particular, the pseudo-dimension of $\mathcal{Z}$ is equal to the dimension of E.*

Note that Lemma 2.2 is very crucial to prove the above theorem. Let $\mathcal{E} = \{E_1, \ldots, E_\gamma\}$ be the error vector set of $\mathcal{Z}$ and let $\{C_1, \ldots, C_\gamma\}$ be the corresponding codewords which **A** sent. Define $BAD = \cup_{i=1}^{\gamma} support(E_i)$. Then BAD is the set of channels that $\mathcal{A}_t$ has corrupted.

Theorem 2.5 ([13])**.** *Let $\mathcal{B}_y = \{Y_{j_1}, \ldots, Y_{j_k}\}$ be the pseudo-basis of $\mathcal{Z}$ and let $C_{j_1}, \ldots, C_{j_k}$ be the corresponding original codewords. Then $BAD = \cup_{i=1}^{k} support(Y_{j_i} - C_{j_i})$. Moreover, the pseudo-dimension of $\mathcal{Z}$ is at most t.*

Having completed the theory of pseudo-basis, it is now easy to contemplate how it could be used in PSMT schemes. Briefly, **A** sends γ random codewords from $\mathcal{C}$ where γ is more than t (which is the bound on pseudo-dimension). **B** receives γ vectors at his end and efficiently computes pseudo-basis ([13] presents an efficient algorithm for the same) and sends it back to **A** reliably. **A**, on receiving the pseudo-basis, finds corresponding error vectors E_{ji} and use $BAD = \cup_{i=1}^{k} support(E_{ji})$ to get the set of all corrupted channels.

3 Pseudo-basis Does not Extend to General Adversary Settings

Here we show that pseudo-basis does not extend for general adversary. We start with general adversary and Linear Secret Sharing schemes (LSSS) and continue our discussion on the equivalence of LSSS to Monotone Span Program (MSP) and Linear Codes for general errors specified by general adversary structure. We also prove several important theorems on LSSS, MSP and Linear Code which will be later utilized to prove our claims.

3.1 General Adversary, Linear Codes, MSPs and LSSSs

Recall that a *general adversary* $\mathcal{A}$ is characterized by an *adversary structure* Γ, which is a collection of subsets of channels that the adversary $\mathcal{A}$ can *potentially* corrupt. That is, $\Gamma = \{B \subset \mathcal{W} \mid \mathcal{A} \text{ can corrupt } B\}$. Assuming the adversary structure to be monotone, we have that if $B \in \Gamma$ and if $B' \subset B$, then $B' \in \Gamma$. We define *access structure* $\Sigma = \Gamma^c$, the complement of Γ. An access structure Σ is monotone if $A \in \Sigma$ and $A' \supseteq A$, then $A' \in \Sigma$.

Linear Secret Sharing schemes (LSSS) and Monotone Span Programme (MSP). In a *secret sharing scheme*, a dealer D distributes a secret $s \in \mathbb{F}$, to a set of N parties $\mathcal{P} = \{P_1, \ldots, P_N\}$ in such a way that only the participants of access sets in access structure can reconstruct s from their shares, while subsets of the participants in adversary structure have no information about s from their shares. A secret sharing scheme for any monotone access structure Σ can be realized by a LSSS [3] as follows: Let $\mathcal{M}$ be a $d \times e$ matrix over $\mathbb{F}$ and $\psi : \{1, \cdots, d\} \rightarrow \{1, \cdots, N\}$ be a labeling function, where $d \geq e$ and $d \geq N$. Also $(1, 0, \cdots, 0)$ is in the linear

span of $\mathcal{M}_A = \{m_j \mid \psi(j) \in A \text{ and } A \in \Sigma\}$, where m_j denotes the jth row of $\mathcal{M}$. So there is a vector $\vec{\alpha}_A$ called *recombination vector*, such that $\vec{\alpha}_A \cdot \mathcal{M}_A = (1, 0, \ldots, 0)$.

Sharing algorithm:

1. To share $s \in \mathbb{F}$, D chooses a random vector $\vec{\rho} \in \mathbb{F}^{e-1}$ and compute a vector $\vec{v} = (v_1, \cdots, v_d)^T = \mathcal{M} \cdot \begin{pmatrix} s \\ \vec{\rho} \end{pmatrix}$.
2. Let $\mathsf{LSSS}(s, \vec{\rho}) = (\mathtt{sh}_1, \cdots, \mathtt{sh}_N)$, where $\mathtt{sh}_i = \{v_j \mid \psi(j) = i\}$. The dealer gives $\mathtt{sh}_i$ to P_i as a share of s for $i = 1, \cdots, N$.

Reconstruction algorithm: The parties in $A \in \Sigma$ reconstruct the secret s as follows: Let $\vec{s}_A$ denote the set of shares corresponding to the parties in A. Then $s = \langle \vec{\alpha}_A, \vec{s}_A^T \rangle$, where $\langle x, y \rangle$ denotes *dot product* of x and y and x^T denotes transpose of x.

In [3], it is shown that the secret s can be reconstructed from the shares of parties in set A if and only if $(1, 0, \cdots, 0)$ is in the linear span of $\mathcal{M}_A = \{m_j \mid \psi(j) \in A\}$. It is well known that LSSSs are equivalent to MSP [3]. We say that the above $(\mathcal{M}, \psi)$ is a monotone span program realizing Σ. The size of the MSP is the number of rows d in M. We now state two important properties of MSP, which will be used later in this paper.

Theorem 3.1. *Two different secrets shared according to an MSP realizing Σ in an* LSSS *cannot have same shares corresponding to an access set.*

Proof. The proof follows from the properties of MSP and is given in **APPENDIX A**. □

Theorem 3.2. *Each of the N shares of a secret generated by an MSP, realizing Σ in an* LSSS *may not be expressed as a linear combination of the shares of the same secret corresponding to an access set. As a special case, we have that even if the shares of some secret corresponding to an access set is all zero, it does not necessarily imply that all the N parties have 0 as their shares.*

Proof. The proof follows from the fact the rows of $\mathcal{M}$ assigned to the parties in the access set may or may not have full rank. If it has full rank, then every other row of $\mathcal{M}$ can be expressed as a linear combination of the rows assigned to the parties in the access set and hence all the N shares of any secret can be linearly expressed by the shares corresponding to the parties in the access set. Otherwise, it may not be possible to do so. □

To illustrate the above theorem, we give the following example. We will again refer to this example when we demonstrate with a concrete example that pseudo-basis fails in non-threshold settings.

Example 3.3 (Illustration of Theorem 3.2)**.** Let $\mathcal{P} = \{P_1, \ldots, P_5\}$ and consider a non-threshold adversary $\mathcal{A}$, specified by adversary structure $\Gamma = \{\{P_1\}, \{P_2\}, \{P_3\}, \{P_4, P_5\}, \{P_1, P_4\}, \{P_1, P_5\}, \{P_2, P_4\}, \{P_2, P_5\}, \{P_3, P_4\}, \{P_3, P_5\}\}$. It is easy to verify that $\mathcal{A}$ satisfies $\mathcal{Q}^2$ condition. The corresponding monotone access structure $\Sigma = \{\{P_1, P_2\}, \{P_1, P_3\}, \{P_2, P_3\}, \{P_1, P_4, P_5\}, \{P_2, P_4, P_5\}, \{P_3, P_4, P_5\}\}$. Now consider the following LSSS: To share a secret s, select a random polynomial $f(x) = s + ax$, where a is random. In addition, select random b, c and d, subject to the condition that $s = b + c + d$. It is easy to note that any set in the adversary structure will have no information about the secret s when the shares corresponding to $P_1, \ldots, P_5$, say $\mathtt{sh}_1, \ldots, \mathtt{sh}_5$ respectively are defined as $\mathtt{sh}_1 = (f(1), b) = (s + a, b)$, $\mathtt{sh}_2 = (f(2), b) = (s + 2a, b)$, $\mathtt{sh}_3 = (f(3), b) = (s + 3a, b)$, $\mathtt{sh}_4 = (c)$ and $\mathtt{sh}_5 = (d)$. Since LSSS's are equivalent to MSP, there exists some MSP realizing the above access structure. Now consider the secret $s = 0$ and randomness $(a, b, c, d) = (0, 0, 1, -1)$. Then the shares of $s = 0$ generated according to the randomness $(a, b, c, d) = (0, 0, 1, -1)$ are: $\mathtt{sh}_1 = (0, 0), \mathtt{sh}_2 = (0, 0), \mathtt{sh}_3 = (0, 0), \mathtt{sh}_4 = (1)$ and $\mathtt{sh}_5 = (-1)$. Now notice that though the shares of the parties corresponding to the access set $\{P_1, P_2, P_3\}$ is $(0, 0, 0, 0, 0, 0)$, it is not the case for the remaining two shares.

Linear Code. In [25], it is shown that every LSSS can be viewed as a *linear error correcting code* [14]. Specifically the following is shown:

Theorem 3.4 ([25])**.** *Let $\mathcal{M}$ be an MSP of size $d \times e$ realizing Σ in an* LSSS*. Let $\mathcal{C}$ be the set of all possible d-length vectors that can be obtained as shares of some secret for some randomness in the* LSSS*. That is $\mathcal{C} = \{y : y \in \mathcal{F}^d$ and $y^T = \mathcal{M} \cdot (s, \vec{\rho})$, for some secret $s \in \mathbb{F}$ and some randomness $\vec{\rho} \in \mathbb{F}^{e-1}\}$. Then $\mathcal{C}$ constitutes a valid linear code.*

In the sequel, we first demonstrate with a concrete example that pseudo-basis does not extend to general adversary and then argue the subtle reason behind it. *Note that in the context of PSMT, the parties in LSSS/MSP can be interpreted as wires.*

3.2 Concrete Demonstration of Non-generalization of Pseudo-basis

In [25] the authors claimed the following: suppose there are N channels from $\mathbf{A}$ to $\mathbf{B}$, denoted by $\mathcal{W} = \{w_1, \ldots, w_N\}$ and let $\mathcal{A}$ be a general adversary, characterized by an adversary structure Γ over $\mathcal{W}$, such that $\mathcal{A}$ satisfies $\mathcal{Q}^2$ (refer to Definition 1.1 for the meaning of $\mathcal{Q}^2$) condition with respect to $\mathcal{W}$ (this is a generalization of the condition $N \geq 2t+1$). Let $\mathcal{M}$ be the MSP realizing the access structure $\Sigma = \Gamma^c$. Moreover, let $\mathcal{C}$ be the linear code corresponding to $\mathcal{M}$ (by Theorem 3.4). Then the concept of pseudo-basis can be extended to general settings. Specifically, Lemma 2.2, Theorem 2.4 and Theorem 2.5 will hold for $\mathcal{A}$. However, we now show that this need not be true. We first present a counter example.

Example 3.5 (Counter Example)**.** We consider Example 3.3 replacing $\mathcal{P} = \{P_1, \ldots, P_5\}$ by $\mathcal{W} = \{w_1, \ldots, w_5\}$. Consider the linear code $\mathcal{C} = \{(s+a, b, s+2a, b, s+3a, b, c, d)\}$ corresponding to the LSSS described in Example 3.3. Here s is the secret and $(a, b, c, d) \in \mathbb{F}^4$ is a random tuple, satisfying the constraints that $b + c + d = s$. Moreover, $\mathtt{sh}_1 = (s + a, b)$, $\mathtt{sh}_2 = (s + 2a, b)$, $\mathtt{sh}_3 = (s + 3a, b)$, $\mathtt{sh}_4 = (c)$ and $\mathtt{sh}_5 = (d)$ are the first, second, third, fourth and fifth component of the codeword.

If $(s, a, b, c, d) = (0, 0, 0, 1, -1)$, then the codeword is $C_1 = (0, 0, 0, 0, 0, 0, 1, -1)$. On the other hand, if $(s, a, b, c, d) = (-1, 1, 0, -1, 1)$, then the codeword is $C_2 = (0, 0, 1, 0, 2, 0, -1, 1)$. Now suppose C_1, C_2 are transmitted by $\mathbf{A}$ over the channels, with $\mathtt{sh}_i$ being transmitted over channel w_i. Let $\mathcal{A}$ controls the set $\{w_4, w_5\}$ from Γ. Let $\mathbf{B}$ receive $Y_1 = (0, 0, 0, 0, 0, 0, 1, 0)$ and $Y_2 = (0, 0, 1, 0, 2, 0, 0, 1)$. So the corresponding error vectors are $E_1 = (0, 0, 0, 0, 0, 0, 0, 1)$ and $E_2 = (0, 0, 0, 0, 0, 0, 1, 0)$.

Now it is easy to see that $Y_2 - Y_1 = (0, 0, 1, 0, 2, 0, -1, 1) = C_2 \in \mathcal{C}$. Thus, Y_1, Y_2 can be linearly pseudo-expressed by Y_1 and hence Y_1 is the pseudo-basis of $\mathcal{Z} = \{Y_1, Y_2\}$. So according to Lemma 2.2 and Theorem 2.4, E_1 should be the basis of error vectors $\mathcal{E} = \{E_1, E_2\}$. However, this is not so as E_1 and E_2 are independent. Now if $\mathbf{B}$ reliably sends back the pseudo-basis Y_1 to $\mathbf{A}$, then from the corresponding error vector $E_1 = Y_1 - C_1$, $\mathbf{A}$ will conclude that only w_5 is corrupted, which is not so as w_4 is also corrupted. □

3.3 The Technicalities for the Non-Applicability of Pseudo-basis

The mean reason that stops pseudo-basis to work in general settings is the invalidity of 'if' part of Lemma 2.2. Specifically, we now prove the following important theorem:

Theorem 3.6. *In general adversary settings, even if any $Y_i \in \mathcal{Z}$ is linearly pseudo-expressed by $\{Y_{j_1}, \ldots, Y_{j_k}\} \subset \mathcal{Z}$, the error vector E_i may not be linearly expressed by error vectors $\{E_{j_1}, \ldots, E_{j_k}\}$.*

Proof. Assume that Y_i be linearly pseudo-expressed by $\{Y_{j_1}, \ldots, Y_{j_k}\}$. Then we have $Y_i = (a_1 Y_{j_1} + \ldots + a_k Y_{j_k})$ *mod* $\mathcal{C}$, for some $a_1, \ldots, a_k$. This implies that:

$$(\mathcal{C}_i + E_i) = \{a_1(\mathcal{C}_{j_1} + E_{j_1}) + \ldots + a_k(\mathcal{C}_{j_k} + E_{j_k})\} \; mod \; \mathcal{C}$$
$$\Rightarrow (\mathcal{C}_i - a_1\mathcal{C}_{j_1} - \ldots - a_k\mathcal{C}_{j_k}) + (E_i - a_1E_{j_1} - \ldots - a_kE_{j_k}) \in \mathcal{C}$$

From linearity property of $\mathcal{C}$, we have $(\mathcal{C}_i - a_1\mathcal{C}_{j_1} - \ldots - a_k\mathcal{C}_{j_k}) \in \mathcal{C}$ and $(E_i - a_1E_{j_1} - \ldots - a_kE_{j_k}) \in \mathcal{C}$. In general adversary settings, each error vector $E_i, E_{j_1}, \ldots, E_{j_k}$ will have 0's in the locations corresponding to honest channels, which forms an access set, say Acc_1 and non-zero values in the locations corresponding to some set of corrupted channels from Γ. This further implies that the codeword $(E_i - a_1E_{j_1} - \ldots - a_kE_{j_k})$ will have 0's in the locations corresponding to Acc_1. But now Theorem 3.2 prohibits us to claim that all the remaining entries of $(E_i - a_1E_{j_1} - \ldots - a_kE_{j_k})$ will also be 0 and hence $(E_i - a_1E_{j_1} - \ldots - a_kE_{j_k})$ is an all 0-codeword (also see Example 3.3 to help the intuition). Thus the fact that $(E_i - a_1E_{j_1} - \ldots - a_kE_{j_k}) = (0, \ldots, 0)$ and hence $E_i = a_1E_{j_1} + \ldots + a_kE_{j_k}$ need not hold in non-threshold settings. □

4 Efficient Three Round PSMT in Directed Network

Let $\mathcal{Y} = \{w_1, \ldots, w_n\}$ be the set of channels in *top band*, directed from $\mathbf{S}$ to $\mathbf{R}$ and $\mathcal{X} = \{q_1, \ldots, q_u\}$ be the set of channels in *bottom band*, directed from $\mathbf{R}$ to $\mathbf{S}$. Let $\mathcal{W} = \mathcal{Y} \cup \mathcal{X}$ and $\mathcal{A}$ be a general adversary specified by an adversary structure Γ over $\mathcal{W}$, such that **N&S-II** is satisfied. We then construct a 3-round PSMT protocol for sending a message $m^{\mathbf{S}}$ containing single (i.e. $\ell = 1$) field element, without relying on pseudo-basis. The basic goal of the protocol is to allow $\mathbf{S}$ and $\mathbf{R}$ to interact with each other to agree on information theoretically secure one time pads, which will be then used to mask the shares of the message. However, there are various subtleties involved

underway, specially due to the reason that $\mathbf{R}$ may not be able to reliably send back any feedback to $\mathbf{S}$ through bottom band, as **N&S-II** does not ensure the bottom band to satisfy $\mathcal{Q}^2$ condition (which is a necessary condition for doing reliable communication [10]). We informally explain the key ideas one by one that help to overcome the hurdles and address the subtleties underway.

Some Important Notations. Let Γ' be the restriction of Γ over the set $\mathcal{Y}$. That is, for each $A \in \Gamma$, there is a corresponding $A' \in \Gamma'$, where $A' = A \cap \mathcal{Y}$. Let $\mathcal{A}'$ be the general adversary, specified by the adversary structure Γ' over $\mathcal{Y}$. Let $\Sigma' = \Gamma'^c$ be the corresponding access structure over $\mathcal{Y}$. Since **N&S-II** is satisfied, $\mathcal{A}$ and therefore $\mathcal{A}'$ satisfies $\mathcal{Q}^2$ condition with respect to $\mathcal{Y}$. Let $\mathcal{M}$ be the $d \times e$ MSP, realizing Γ'. *Without loss of generality and for the sake of simplicity, we assume that $d = n$ and i^{th} row of $\mathcal{M}$ is assigned to w_i, for $i = 1, \ldots, n$.* A channel which is under the control of the adversary is called *corrupted*. The rest of the wires are terms as *honest*. We say $\mathbf{S}$ broadcasts some value to $\mathbf{R}$, if it sends the value over all the wires in top band. Similarly, $\mathbf{R}$ broadcasts some value when it sends the value over all wires in bottom band.

The First Step. With the goal to establish a set of information theoretically secure random pads with $\mathbf{R}$, let $\mathbf{S}$ do the following: He selects n random secrets, say $s_1^{\mathbf{S}}, \ldots, s_n^{\mathbf{S}}$ and computes n shares for each secret $s_i^{\mathbf{S}}$ as $(\mathtt{sh}_{i,1}^{\mathbf{S}}, \cdots, \mathtt{sh}_{i,n}^{\mathbf{S}}) = \mathsf{LSSS}(s_i^{\mathbf{S}}, \vec{\rho_i}^{\mathbf{S}})$, where $\vec{\rho_i}^{\mathbf{S}} \in \mathbb{F}^{e-1}$ is a random vector. The n secrets and their n shares are considered as a single block of information $B^{\mathbf{S}}$ that $\mathbf{S}$ communicates to $\mathbf{R}$ over $\mathcal{Y}$ as follows: $(s_i^{\mathbf{S}}, \vec{\rho_i}^{\mathbf{S}})$ and i^{th} share of all the n secrets, namely $\mathtt{sh}_{j,i}^{\mathbf{S}}$, for $j = 1, \ldots, n$ are sent over channel w_i, for $i = 1, \ldots, n$.

Let $\mathbf{R}$ receive $B^{\mathbf{R}}$ that contains $(s_i^{\mathbf{R}}, \vec{\rho_i}^{\mathbf{R}})$ and the shares $\mathtt{sh}_{j,i}^{\mathbf{R}}$ over channel w_i for $i = 1, \ldots, n$. $\mathbf{R}$ finds inconsistencies among the received values as follows: $\mathbf{R}$ computes the n shares for each $s_i^{\mathbf{R}}$ himself as $(\mathtt{sh}'^{\mathbf{R}}_{i,1}, \ldots, \mathtt{sh}'^{\mathbf{R}}_{i,n}) = \mathsf{LSSS}(s_i^{\mathbf{R}}, \vec{\rho_i}^{\mathbf{R}})$. He then verifies $(\mathtt{sh}'^{\mathbf{R}}_{i,1}, \ldots, \mathtt{sh}'^{\mathbf{R}}_{i,n}) \stackrel{?}{=} (\mathtt{sh}_{i,1}^{\mathbf{R}}, \ldots, \mathtt{sh}_{i,n}^{\mathbf{R}})$, which should ideally hold. If there are some $j \in \{1, \ldots, n\}$, such that $\mathtt{sh}'^{\mathbf{R}}_{i,j} \neq \mathtt{sh}_{i,j}^{\mathbf{R}}$ then $\mathbf{R}$ knows that either channel w_i is corrupted and $(s_i^{\mathbf{R}}, \vec{\rho_i}^{\mathbf{R}}) \neq (s_i^{\mathbf{S}}, \vec{\rho_i}^{\mathbf{S}})$ or w_j is corrupted and $\mathtt{sh}_{i,j}^{\mathbf{R}} \neq \mathtt{sh}_{i,j}^{\mathbf{S}}$. But $\mathbf{R}$ is not capable of identifying the exact corrupted channel. So $\mathbf{R}$ adds a *conflicting tuple* $(i, j, \mathtt{sh}'^{\mathbf{R}}_{i,j}, \mathtt{sh}_{i,j}^{\mathbf{R}})$ in a *conflict list* $\mathsf{CL}^{\mathbf{R}}$. We now claim some important properties of $\mathsf{CL}^{\mathbf{R}}$.

Claim 4.1. (1) $|\mathsf{CL}^{\mathbf{R}}| \leq n^2$; (2) If w_i is corrupted and delivered $s_i^{\mathbf{R}} \neq s_i^{\mathbf{S}}$, then there is at least one honest channel, say w_j, such that the conflicting tuple $(i, j, \mathtt{sh}'^{\mathbf{R}}_{i,j}, \mathtt{sh}_{i,j}^{\mathbf{R}})$ will be present in $\mathsf{CL}^{\mathbf{R}}$; (3) If w_i and w_j are honest, then there will be no tuple of the form $(i, j, \star, \star)$ and $(j, i, \star, \star)$ in $\mathsf{CL}^{\mathbf{R}}$; (4) If w_i is honest then $\mathcal{A}$ will have no information about $s_i^{\mathbf{S}}$. Moreover, if there is a tuple $(i, j, \mathtt{sh}'^{\mathbf{R}}_{i,j}, \mathtt{sh}_{i,j}^{\mathbf{R}})$ in $\mathsf{CL}^{\mathbf{R}}$, then w_j is corrupted and $\mathcal{A}$ already knows $\mathtt{sh}'^{\mathbf{R}}_{i,j} = \mathtt{sh}_{i,j}^{\mathbf{S}} \neq \mathtt{sh}_{i,j}^{\mathbf{R}}$.

Proof: The first part is obvious. We prove the second part by contradiction. Suppose there is no such honest channel w_j, which will conflict with w_i. This implies that two different secrets $s_i^{\mathbf{R}} \neq s_i^{\mathbf{S}}$ have same shares corresponding to all honest channels (which form an access set). However, from Theorem 3.1 this is not possible.

Part (3) of the claim is obvious. Since w_i is an honest channel, the adversary can not listen $(s_i^{\mathbf{S}}, \vec{\rho_i}^{\mathbf{S}})$ directly from w_i. However, adversary will know the shares of $s_i^{\mathbf{S}}$, namely $\mathtt{sh}_{i,j}^{\mathbf{S}}$, if w_j is under the control of him. Unfortunately, the channels under the control of the adversary do not constitute an access set and so from the properties of MSP, $s_i^{\mathbf{S}}$ remains hidden from adversary in information theoretic sense. The rest of the claim is obvious. □

The second assertion above suggests that $\mathbf{S}$ can identify all corrupted channels w_i, that delivered incorrect $s_i^{\mathbf{R}}$'s to $\mathbf{R}$, provided $\mathsf{CL}^{\mathbf{R}}$. But reliable communication of $\mathsf{CL}^{\mathbf{R}}$ through the bottom band is ruled out since the bottom band $\mathcal{X}$ may not satisfy $\mathcal{Q}^2$. [2] Of course, $\mathbf{R}$ can broadcast $\mathsf{CL}^{\mathbf{R}}$ and wait for the feedback from $\mathbf{S}$ based on which $\mathbf{R}$ can verify whether $\mathbf{S}$ correctly received $\mathsf{CL}^{\mathbf{R}}$ or not. In fact this approach was followed in the existing PSMT protocols of [5, 15, 20]. Apparently there are two main issues to be addressed now: (a) $\mathbf{S}$ might receive a bunch of different conflict lists though the bottom band and now without even knowing whether the correct conflict list is received or which of the received conflict list is the actual one, $\mathbf{S}$ should manage to establish the common pad and communicate message in the third round. (b) $\mathbf{S}$ also needs to protect his message by taking counter measure against *guessing attack* [24]. The existing protocols of [5, 15, 20] fall in the trap of *guessing attack*. Our protocol resolves the above issues in a clever way, as discussed below. Before that we give a brief account on guessing attack.

[2] From [10], $\mathcal{Q}^2$ condition is necessary for doing reliable communication. If the channels from $\mathbf{A}$ to $\mathbf{B}$ satisfy $\mathcal{Q}^2$ condition, then $\mathbf{A}$ can reliably send any value by **broadcasting** it (i.e., sending it over all the channels). $\mathbf{B}$ can then recover the value efficiently by using the method given in [12].

Guessing Attack. Consider the following steps: **R broadcasts** $\mathsf{CL}^{\mathbf{R}}$. **S** may receive a bunch of different conflict lists. **S** may not know whether the received list(s) contain $\mathsf{CL}^{\mathbf{R}}$ or which of the received lists is the correct one. So considering each received list as a valid one, **S** finds the list of corrupted channels corresponding each such list. **S** then broadcasts all conflict lists and their corresponding corrupted channel lists.

Now a guessing attack in any protocol that have above steps is mounted as follows: In the worst case, the entire $\mathcal{X}$ may be under the control of the adversary. For each wire in the bottom band, adversary then makes random *guesses* on the shares of secrets $s_i^{\mathbf{S}}$s corresponding to honest w_is, constructs fake conflict lists each containing a number of conflictions between honest and corrupted channels and delivers this fake list in the place of actual list. Having done this, he waits for the feedback of **S** in the next round based on these fake conflict lists. Now if any of the fake conflictions indeed happened to be a genuine confliction (implying adversary's guess is correct), then the adversary gets a correct share (which was guessed by the adversary to generate the confliction) of the a secret which was earlier unknown to the adversary. On the other hand, if his guess of the value for the honest wire in a confliction is wrong, he gets to know it and thereby is able to prune the search space for the unknown shares of an unknown secret. This clearly violates the *perfect secrecy* of some secrets which would remain secret if **S** would not have send the feedbacks. Essentially, the adversary is getting access to an oracle service from **S** confirming about whether a number of different random guesses on the unknown shares of a secret are correct or not. For details, see [24].

Round I of Our Protocol (Avoiding Guessing Attack by Union Technique). As discussed previously, the adversary may mount guessing attack and may be able to break perfect secrecy of all the secrets in block $B^{\mathbf{S}}$ sent by **S** to **R** in the hope of establishing common information theoretic pads. We address the problem by making **S** to send $z = n^2u + 1$ blocks of secrets instead of a single block $B^{\mathbf{S}}$ and then making **R** to construct and broadcast a single compressed conflict list for all the blocks instead of individual conflict lists for each of the z blocks. Recall that a single block contains n random secrets and n shares for each of them. The combined conflict list may contain at most n^2 conflictions (possibly) each from different blocks. The adversary, having control over all the u wires in the bottom band (in worst case) can mount guessing attacks on the secrets of at most n^2u blocks. Since $z = n^2u + 1$, there is still one block of secrets where the adversary gets no scope to mount guessing attack and (some part of) these secrets will act as the information theoretic random pads between **S** and **R**. Now the flow of information for $z = n^2u + 1$ blocks over $\mathcal{Y}$ will be as follows:

Figure 1: Data Flow Over the n Channels. Here $z = n^2u + 1$

Channel	$B_1^{\mathbf{S}}$	...	$B_k^{\mathbf{S}}$	...	$B_z^{\mathbf{S}}$
w_1	$(s_{11}^{\mathbf{S}}, \vec{\rho_{11}}^{\mathbf{S}})$ $\mathtt{sh}_{1,1,1}^{\mathbf{S}}, \ldots, \mathtt{sh}_{1,n,1}^{\mathbf{S}}$	... ...	$(s_{k1}^{\mathbf{S}}, \vec{\rho_{k1}}^{\mathbf{S}})$ $\mathtt{sh}_{k,1,1}^{\mathbf{S}}, \ldots, \mathtt{sh}_{k,n,1}^{\mathbf{S}}$	... ...	$(s_{z1}^{\mathbf{S}}, \vec{\rho_{z1}}^{\mathbf{S}})$ $\mathtt{sh}_{z,1,1}^{\mathbf{S}}, \ldots, \mathtt{sh}_{z,n,1}^{\mathbf{S}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
w_i	$(s_{1i}^{\mathbf{S}}, \vec{\rho_{1i}}^{\mathbf{S}})$ $\mathtt{sh}_{1,1,i}^{\mathbf{S}}, \ldots, \mathtt{sh}_{1,n,i}^{\mathbf{S}}$	... ...	$(s_{ki}^{\mathbf{S}}, \vec{\rho_{ki}}^{\mathbf{S}})$ $\mathtt{sh}_{k,1,i}^{\mathbf{S}}, \ldots, \mathtt{sh}_{k,n,i}^{\mathbf{S}}$	... ...	$(s_{zi}^{\mathbf{S}}, \vec{\rho_{zi}}^{\mathbf{S}})$ $\mathtt{sh}_{z,1,i}^{\mathbf{S}}, \ldots, \mathtt{sh}_{z,n,i}^{\mathbf{S}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
w_n	$(s_{1n}^{\mathbf{S}}, \vec{\rho_{1n}}^{\mathbf{S}})$ $\mathtt{sh}_{1,1,n}^{\mathbf{S}}, \ldots, \mathtt{sh}_{1,n,n}^{\mathbf{S}}$	... ...	$(s_{kn}^{\mathbf{S}}, \vec{\rho_{kn}}^{\mathbf{S}})$ $\mathtt{sh}_{k,1,n}^{\mathbf{S}}, \ldots, \mathtt{sh}_{k,n,n}^{\mathbf{S}}$	... ...	$(s_{zn}^{\mathbf{S}}, \vec{\rho_{zn}}^{\mathbf{S}})$ $\mathtt{sh}_{z,1,n}^{\mathbf{S}}, \ldots, \mathtt{sh}_{z,n,n}^{\mathbf{S}}$

Corresponding to $B_k^{\mathbf{S}}$, **R** receives $B_k^{\mathbf{R}}$, consisting of pairs $(s_{ki}^{\mathbf{R}}, \vec{\rho_{ki}}^{\mathbf{R}})$ and shares $\mathtt{sh}_{k,i,j}^{\mathbf{R}}$. As described before, **R** constructs conflict lists $\mathsf{CL}_k^{\mathbf{R}}$ for every block $B_k^{\mathbf{R}}$ consisting of the conflicting tuples $(i, j, \mathtt{sh'}_{k,i,j}^{\mathbf{R}}, \mathtt{sh}_{k,i,j}^{\mathbf{R}})$, such that $\mathtt{sh'}_{k,i,j}^{\mathbf{R}} \neq \mathtt{sh}_{k,i,j}^{\mathbf{R}}$, where $(\mathtt{sh'}_{k,i,1}^{\mathbf{R}}, \ldots, \mathtt{sh'}_{k,i,n}^{\mathbf{R}}) = \mathsf{LSSS}(s_{ki}^{\mathbf{R}}, \vec{\rho_{ki}}^{\mathbf{R}})$. Now, Claim 4.1 holds for every conflict list $\mathsf{CL}_k^{\mathbf{R}}$.

Idea of Union Technique: The idea is to construct a *single combined conflict list* $\mathsf{CCL}^{\mathbf{R}}$ from n^2u+1 individual conflict lists by taking their union, recording confliction between w_i and w_j *only once*, from the lowest indexed conflict list containing a confliction between w_i and w_j. Specifically, if there is a confliction of the form $(i, j, \star, \star)$ in at least one $\mathsf{CL}_k^{\mathbf{R}}$, then **R** puts a 5-tuple $(\alpha, i, j, \mathtt{sh'}_{\alpha,i,j}^{\mathbf{R}}, \mathtt{sh}_{\alpha,i,j}^{\mathbf{R}})$ in $\mathsf{CCL}^{\mathbf{R}}$ where $\alpha = \min\{1, \ldots, z\}$ such that $\mathsf{CL}_\alpha^{\mathbf{R}}$ contains tuple $(i, j, \mathtt{sh'}_{\alpha,i,j}^{\mathbf{R}}, \mathtt{sh}_{\alpha,i,j}^{\mathbf{R}})$. Now we claim the following:

Claim 4.2 (Properties of Combined Conflict List)**.** (1) $|\mathsf{CCL}^{\mathbf{R}}| \leq n^2$; (2) If a corrupted w_i delivered $s_{ki}^{\mathbf{R}} \neq s_{ki}^{\mathbf{S}}$, $k \in \{1, \ldots, n^2u + 1\}$, then there is at least one honest channel, say w_j, such that there exists a tuple $(\gamma, i, j, \mathtt{sh'}_{\gamma,i,j}^{\mathbf{R}}, \mathtt{sh}_{\gamma,i,j}^{\mathbf{R}}) \in \mathsf{CCL}^{\mathbf{R}}$, where $1 \leq \gamma \leq k$.

Proof. The first part follows from the fact that there can be at most n^2 conflictions involving two channels out of n channels and only one instance of confliction between w_i and w_j is recorded in $\mathsf{CCL}^{\mathbf{R}}$.

Since Claim 4.1 holds for every $\mathsf{CL}_k^{\mathbf{R}}$, from part(2) of the claim, if w_i has delivered $s_{ki}^{\mathbf{R}} \neq s_{ki}^{\mathbf{S}}$, where $k \in \{1, \ldots, n^2u+1\}$, then there exists at least one honest channel, say w_j, such that the conflicting tuple $(i, j, \mathtt{sh}'^{\mathbf{R}}_{k,i,j}, \mathtt{sh}^{\mathbf{R}}_{k,i,j})$ will be present in the kth conflict list $\mathsf{CL}_k^{\mathbf{R}}$. Now the second part of the claim follows from the fact that combined conflict list $\mathsf{CCL}^{\mathbf{R}}$ is constructed by $\mathbf{R}$ by taking union of all n^2u+1 individual conflict lists, where a confliction is recorded only once in the combined list, from the lowest indexed conflict list containing the confliction. So if the confliction between corrupted w_i and honest w_j has not occurred in any $B_\gamma^{\mathbf{R}}$, where $\gamma < k$, then the confliction $(k, i, j, \mathtt{sh}'^{\mathbf{R}}_{k,i,j}, \mathtt{sh}^{\mathbf{R}}_{k,i,j})$ will be present $\mathsf{CCL}^{\mathbf{R}}$. On the other hand, if confliction between corrupted w_i and honest w_j has also occurred in some $B_\gamma^{\mathbf{R}}$, where $\gamma < k$, then the confliction $(\gamma, i, j, \mathtt{sh}'^{\mathbf{R}}_{\gamma,i,j}, \mathtt{sh}^{\mathbf{R}}_{\gamma,i,j})$ will be present $\mathsf{CCL}^{\mathbf{R}}$. □

Round II of the Protocol. During second round, **R broadcasts** $\mathsf{CCL}^{\mathbf{R}}$. *We consider the worst case where* **S** *receives* u *distinct lists of* 5*-tuples, denoted by* $CCL_1^{\mathbf{S}}, \ldots, CCL_u^{\mathbf{S}}$ *through* $q_1, \ldots, q_u$ *respectively.* Since **S** has no way of knowing whether any of the received lists is same as $\mathsf{CCL}^{\mathbf{R}}$, he considers each received list as valid and finds the list of corrupted channels corresponding to it. So **S** constructs *corrupted channel lists* $\mathsf{CChL}_1^{\mathbf{S}}, \ldots, \mathsf{CChL}_u^{\mathbf{S}}$ as follows: For each $p \in \{1, \ldots, u\}$, **S** considers each conflicting 5-tuple $(\alpha, i, j, \mathtt{sh}'^{\mathbf{R}}_{\alpha,i,j}, \mathtt{sh}^{\mathbf{R}}_{\alpha,i,j}) \in \mathsf{CCL}_p^{\mathbf{S}}$, and finds out whether w_i or w_j is corrupted by checking $\mathtt{sh}'^{\mathbf{R}}_{\alpha,i,j} \stackrel{?}{=} \mathtt{sh}^{\mathbf{S}}_{\alpha,i,j}$ and $\mathtt{sh}^{\mathbf{R}}_{\alpha,i,j} \stackrel{?}{=} \mathtt{sh}^{\mathbf{S}}_{\alpha,i,j}$. If the first test fails then **S** adds corrupted w_i to $\mathsf{CChL}_p^{\mathbf{S}}$, since w_i delivered $(s_{\alpha i}^{\mathbf{R}}, \vec{\rho_{\alpha i}}^{\mathbf{R}}) \neq (s_{\alpha i}^{\mathbf{S}}, \vec{\rho_{\alpha i}}^{\mathbf{S}})$. On the other hand, if the second test fails, then **S** adds corrupted w_j to $\mathsf{CChL}_p^{\mathbf{S}}$, since w_j delivered $\mathtt{sh}^{\mathbf{R}}_{\alpha,i,j} \neq \mathtt{sh}^{\mathbf{S}}_{\alpha,i,j}$. Now here **S** performs a crucial step. If $|\mathsf{CCL}_p^{\mathbf{S}}| > n^2$, for any $p \in \{1, \ldots, u\}$, then **S** ignores $\mathsf{CCL}_p^{\mathbf{S}}$ and concludes q_p to be faulty since from part(1) of Claim 4.2, $|\mathsf{CCL}^{\mathbf{R}}| \leq n^2$. Now we have the following:

Claim 4.3. There is at least one $k \in \{1, \ldots, n^2u+1\}$, such that no $\mathsf{CCL}_p^{\mathbf{S}}$ (that is not ignored by **S**) contains a 5-tuple of the form $(k, \star, \star, \star, \star)$.

Proof. The proof follows from the fact that the first component of any 5-tuple can take value from the set $\{1, \ldots, n^2u+1\}$ and each $\mathsf{CCL}_p^{\mathbf{S}}$ which is not ignored by **S** satisfies $|\mathsf{CCL}_p^{\mathbf{S}}| \leq n^2$. □

Round III of the Protocol. Let δ be the minimum index in $\{1, \ldots, n^2u+1\}$, such that there exists no tuple of the form $(\delta, \star, \star, \star, \star)$ in any of the non-discarded $\mathsf{CCL}_1^{\mathbf{S}}, \ldots, \mathsf{CCL}_u^{\mathbf{S}}$ (Claim 4.3 guarantees the existence of such δ). This implies that there is no conflicting tuple present in any $\mathsf{CCL}_1^{\mathbf{S}}, \ldots, \mathsf{CCL}_u^{\mathbf{S}}$, involving values from the block $B_\delta^{\mathbf{R}}$. **S** uses the secrets in the block $B_\delta^{\mathbf{S}}$ as pads. Specifically, **S** computes n shares for message $m^{\mathbf{S}}$ and masks the ith share $\mathtt{sh}_i^{\mathbf{S}}$ of $m^{\mathbf{S}}$ with the ith secret of block $B_\delta^{\mathbf{S}}$, namely $s_{\delta i}^{\mathbf{S}}$. Finally, **S broadcasts** each received $\mathsf{CCL}_p^{\mathbf{S}}$, if it is not ignored by **S** and corresponding $\mathsf{CChL}_p^{\mathbf{S}}$. In addition, **S** also broadcasts the index δ and masked shares of $m^{\mathbf{S}}$. Notice that all these values will be received correctly by **R**, as $\mathcal{Y}$ satisfies $\mathcal{Q}^2$. Now guessing attack cannot be mounted because **S** is using the secrets of block $B_\delta^{\mathbf{S}}$ as pads and the feedback of **S** during third round does not give any information about $B_\delta^{\mathbf{S}}$. So from part(4) of Claim 4.1, $\mathcal{A}$ will not get any information about $s_{\delta i}^{\mathbf{S}}$'s, transmitted over honest w_i's and hence the corresponding masked shares.

Message Recovery by R. If **R** finds that the actual conflict list $\mathsf{CCL}^{\mathbf{R}}$ is one of the conflict lists $\mathsf{CCL}_p^{\mathbf{S}}$ broadcasted by **S**, then message recovery is comparatively easier than the other case where none of the conflict lists $\mathsf{CCL}_p^{\mathbf{S}}$ are equal to $\mathsf{CCL}^{\mathbf{R}}$. In the former case, **R**, knowing the actual corrupted channel list, ignores all $s_{\delta i}^{\mathbf{R}}$'s received over those channels. Using the remaining $s_{\delta i}^{\mathbf{R}}$'s, he then unmasks the corresponding shares of $m^{\mathbf{S}}$. Since these shares will correspond to an access set, **R** can get back $m^{\mathbf{S}}$ from them.

In the later case, the only information **R** knows from **S**'s feedback is that entire $\mathcal{X}$ is corrupted. So message recovery can be done if **R** can find the set $A \in \Gamma$ which got corrupted during the protocol. Notice that here now **R** have some information about the A, namely that $\mathcal{X} \subseteq A$. Unfortunately, there can be several A's in Γ, such that $\mathcal{X} \subseteq A$. To deal with this problem, we use an idea from [11]. Specifically **R** does the following: let

$$\Gamma_{top} = \{A \cap \mathcal{W} \mid A \in \Gamma,\ A \supseteq \mathcal{X}\}.$$

Thus Γ_{top} is the restriction of the sets from Γ which covers the bottom band, to the top band. For a vector $\vec{e} = (e_1, \ldots, e_n)$, let $support(\vec{e}) = \{i \mid e_i \neq 0\}$. Now **R** considers a secret $s_{\delta i}^{\mathbf{R}}$ in block $B_\delta^{\mathbf{R}}$ as *valid*, if $(\mathtt{sh}'^{\mathbf{R}}_{\delta,i,1}, \ldots, \mathtt{sh}'^{\mathbf{R}}_{\delta,i,n})$ and the $(\mathtt{sh}^{\mathbf{R}}_{\delta,i,1}, \ldots, \mathtt{sh}^{\mathbf{R}}_{\delta,i,n})$ differs in the channels, corresponding to some set in Γ_{top}. *Notice that this checking*

requires computation, which is polynomial in the size of MSP because we just have to verify whether target vector lies in the span of the rows which are assigned to these channels (in which the above shares differ) in $\mathcal{M}$. We will show that all the valid secrets are indeed the correct secrets (otherwise second condition of **N&S-II** is violated) and the wires bearing them constitute an access set. So using these secrets, $\mathbf{R}$ can unmask the corresponding shares of $m^{\mathbf{S}}$ and recover $m^{\mathbf{S}}$ from them. In the sequel, we present reliability proof for each case. The complete protocol is now as follows:

Round I: S to R: $\mathbf{S}$ sends blocks of values $B_1^{\mathbf{S}}, \ldots, B_z^{\mathbf{S}}$ as shown in Fig. 1, where each block $B_k^{\mathbf{S}}$ contains the secrets $s_{ki}^{\mathbf{S}}$, for $i = 1, \ldots, n$ and n shares for each $s_{kj}^{\mathbf{S}}$, for $j = 1, \ldots, n$, denoted by $\mathtt{sh}_{k,j,i}^{\mathbf{S}}$.

Round II: R to S: $\mathbf{R}$ receives blocks $B_1^{\mathbf{R}}, \ldots, B_z^{\mathbf{R}}$ and constructs conflict lists $\mathsf{CL}_1^{\mathbf{R}}, \ldots, \mathsf{CL}_z^{\mathbf{R}}$ from $B_1^{\mathbf{R}}, \ldots, B_z^{\mathbf{R}}$ respectively. From these z conflict lists, $\mathbf{R}$ constructs a single combined conflict list $\mathsf{CCL}^{\mathbf{R}}$ by applying union technique. $\mathbf{R}$ then broadcasts $\mathsf{CCL}^{\mathbf{R}}$ to $\mathbf{S}$ through $\mathcal{X}$.

Round III: S to R:

1. Let $\mathbf{S}$ receive $\mathsf{CCL}_p^{\mathbf{S}}$ over channel q_p, for $p = 1, \ldots, u$ If $|\mathsf{CCL}_p^{\mathbf{S}}| > n^2$, then $\mathbf{S}$ ignores/discards $\mathsf{CCL}_p^{\mathbf{S}}$. $\mathbf{S}$ computes the corrupted channel list $\mathsf{CChL}_p^{\mathbf{S}}$ for every non-discarded $\mathsf{CCL}_p^{\mathbf{S}}$.
2. $\mathbf{S}$ computes $(\mathtt{sh}_1^{\mathbf{S}}, \cdots, \mathtt{sh}_n^{\mathbf{S}}) = \mathsf{LSSS}(m^{\mathbf{S}}, \vec{\rho}^{\mathbf{S}})$, where $\vec{\rho}^{\mathbf{S}} \in \mathbb{F}^{e-1}$ is some random vector. Let δ be the least index from $\{1, \ldots, n^2u + 1\}$, such that there exists no 5-tuple of the form $(\delta, \star, \star, \star, \star)$ in any non-discarded $\mathsf{CCL}_p^{\mathbf{S}}$. $\mathbf{S}$ then computes $c_i^{\mathbf{S}}$, where $c_i^{\mathbf{S}} = \mathtt{sh}_i^{\mathbf{S}} \oplus s_{\delta i}^{\mathbf{S}}$, for $i = 1, \ldots, n$.
3. $\mathbf{S}$ broadcasts the index δ, the pairs $(\mathsf{CCL}_p^{\mathbf{S}}, \mathsf{CChL}_p^{\mathbf{S}})$ for every non-discarded $\mathsf{CCL}_p^{\mathbf{S}}$ and the masked shares $c_i^{\mathbf{S}}$, for $i = 1, \ldots, n$.

Message Recovery by R:

1. $\mathbf{R}$ receives index δ, the pairs $(\mathsf{CCL}_p^{\mathbf{S}}, \mathsf{CChL}_p^{\mathbf{S}})$ and masked shares $c_i^{\mathbf{S}}$.
2. If there is a pair $(\mathsf{CCL}_\alpha^{\mathbf{S}}, \mathsf{CChL}_\alpha^{\mathbf{S}})$ such that $\mathsf{CCL}_\alpha^{\mathbf{S}} = \mathsf{CCL}^{\mathbf{R}}$, then $\mathbf{R}$ computes $\mathtt{sh}_i^{\mathbf{R}} = c_i^{\mathbf{S}} \oplus s_{\delta i}^{\mathbf{R}}$ for each $w_i \notin \mathsf{CChL}_\alpha^{\mathbf{S}}$, applies reconstruction algorithm of LSSS to the $\mathtt{sh}_i^{\mathbf{R}}$'s and reconstruct $m^{\mathbf{R}}$.
3. If there is no $(\mathsf{CCL}_\alpha^{\mathbf{S}}, \mathsf{CChL}_\alpha^{\mathbf{S}})$ where $\mathsf{CCL}_\alpha^{\mathbf{S}} = \mathsf{CCL}^{\mathbf{R}}$, then $\mathbf{R}$ computes the following:

$$\mathtt{sh}_i^{\mathbf{R}} = \begin{cases} c_i^{\mathbf{S}} \oplus s_{\delta i}^{\mathbf{R}} & \text{if } support((\mathtt{sh'}_{\delta,i,1}^{\mathbf{R}}, \ldots, \mathtt{sh'}_{\delta,i,n}^{\mathbf{R}}) - (\mathtt{sh}_{\delta,i,1}^{\mathbf{R}}, \ldots, \mathtt{sh}_{\delta,i,n}^{\mathbf{R}})) \in \Gamma_{top}, \\ \perp & \text{otherwise} \end{cases}$$

Here $(\mathtt{sh'}_{\delta,i,1}^{\mathbf{R}}, \ldots, \mathtt{sh'}_{\delta,i,n}^{\mathbf{R}}) = \mathsf{LSSS}(s_{\delta i}^{\mathbf{R}}, \vec{\rho_{\delta i}}^{\mathbf{R}})$. Finally, $\mathbf{R}$ applies reconstruction algorithm of the LSSS to $\mathtt{sh}_i^{\mathbf{R}}$s such that $\mathtt{sh}_i^{\mathbf{R}} \neq \perp$ and reconstructs $m^{\mathbf{R}}$.

Claim 4.4. If $\mathbf{S}$ receives $\mathsf{CCL}^{\mathbf{R}}$ through some channel in $\mathcal{X}$, then at the end of **Round III**, $\mathbf{R}$ will identify all corrupted channels w_i that delivered at least one $s_{ki}^{\mathbf{R}} \neq s_{ki}^{\mathbf{S}}$, where $k \in \{1, \ldots, n^2u + 1\}$. If $\mathbf{S}$ has not received the original combined conflict list, then at the end of **Round III** $\mathbf{R}$ knows all channels in $\mathcal{X}$ are corrupted.

Proof. From Claim 4.2, if a corrupted w_i delivers some $s_{ki}^{\mathbf{R}} \neq s_{ki}^{\mathbf{S}}$ where $k \in \{1, \ldots, n^2u + 1\}$ then there will be a confliction of the form $(\gamma, i, j, \mathtt{sh'}_{\gamma,i,j}^{\mathbf{R}}, \mathtt{sh}_{\gamma,i,j}^{\mathbf{R}})$ in the combined conflict list $\mathsf{CCL}^{\mathbf{R}}$ for some γ in $\{1, \ldots, k\}$ and honest channel w_j. Now if $\mathbf{S}$ receives $\mathsf{CCL}^{\mathbf{R}}$ through channel q_α for some $\alpha \in \{1, \ldots, u\}$, then we have that $\mathsf{CCL}_\alpha^{\mathbf{S}} = \mathsf{CCL}^{\mathbf{R}}$. When $\mathbf{S}$ considers $(\gamma, i, j, \mathtt{sh'}_{\gamma,i,j}^{\mathbf{R}}, \mathtt{sh}_{\gamma,i,j}^{\mathbf{R}}) \in \mathsf{CCL}_\alpha^{\mathbf{S}}$, he finds that $\mathtt{sh'}_{\gamma,i,j}^{\mathbf{R}} \neq \mathtt{sh}_{\gamma,i,j}^{\mathbf{S}}$ and $\mathtt{sh}_{\gamma,i,j}^{\mathbf{R}} = \mathtt{sh}_{\gamma,i,j}^{\mathbf{S}}$. Thus $\mathbf{S}$ will include w_i in $\mathsf{CChL}_\alpha^{\mathbf{S}}$. Since **Round III** communication is reliable, $\mathbf{R}$ receives $(\mathsf{CCL}_\alpha^{\mathbf{S}}, \mathsf{CChL}_\alpha^{\mathbf{S}})$ with $\mathsf{CCL}_\alpha^{\mathbf{S}} = \mathsf{CCL}^{\mathbf{R}}$ and therefore identifies every w_i in $\mathsf{CChL}_\alpha^{\mathbf{S}}$ to be corrupted. The rest of the claim is obvious. □

Lemma 4.5 (Reliability-I)**.** *If $\mathbf{S}$ receives original combined conflict list $\mathsf{CCL}^{\mathbf{R}}$ correctly through at least one channel in $\mathcal{X}$, then $\mathbf{R}$ will be able to correctly recover $m^{\mathbf{S}}$.*

Proof. By Claim 4.4, if $\mathbf{S}$ receives the original combined conflict list $\mathsf{CCL}^{\mathbf{R}}$, then at the end of **Round III**, $\mathbf{R}$ will identify all corrupted channels w_i that delivered $s_{\delta i}^{\mathbf{R}} \neq s_{\delta i}^{\mathbf{S}}$. For the remaining w_i's, we have $s_{\delta i}^{\mathbf{R}} = s_{\delta i}^{\mathbf{S}}$ and furthermore these $s_{\delta i}^{\mathbf{R}}$'s will correspond to an access set. Since all masked shares $c_i^{\mathbf{S}}$'s are broadcasted, they will be received correctly by $\mathbf{R}$. So using the correct $s_{\delta i}^{\mathbf{R}}$'s, $\mathbf{R}$ computes $\mathtt{sh}_i^{\mathbf{R}} = c_i^{\mathbf{S}} \oplus s_{\delta i}^{\mathbf{R}}$, where $\mathtt{sh}_i^{\mathbf{R}}$ is same as $\mathtt{sh}_i^{\mathbf{S}}$. So by applying reconstruction algorithm of LSSS to these $\mathtt{sh}_i^{\mathbf{R}}$'s, $\mathbf{R}$ will correctly recover $m^{\mathbf{R}} = m^{\mathbf{S}}$. □

Lemma 4.6 (Reliability-II)**.** *Even if $\mathbf{S}$ does not receive $\mathsf{CCL}^{\mathbf{R}}$ through any channel in $\mathcal{X}$, $\mathbf{R}$ will recover $m^{\mathbf{S}}$.*

Proof. If $\mathbf{S}$ does not receive $\mathsf{CCL}^{\mathbf{R}}$ then the entire bottom band is corrupted and $\mathbf{R}$ will know this at the end of **Round III** by looking at the set of conflict lists broadcasted by $\mathbf{S}$. Now there are two possible cases:

1. $w_i \in \mathcal{Y}$ *is honest*: In this case $s^{\mathbf{R}}_{\delta i} = s^{\mathbf{S}}_{\delta i}$ and indeed $support((\mathtt{sh'}^{\mathbf{R}}_{\delta,i,1}, \ldots, \mathtt{sh'}^{\mathbf{R}}_{\delta,i,n}) - (\mathtt{sh}^{\mathbf{R}}_{\delta,i,1}, \ldots, \mathtt{sh}^{\mathbf{R}}_{\delta,i,n})) \in \Gamma_{top}$, as no honest channels in $\mathcal{Y}$ will conflict with w_i. Thus, $\mathtt{sh}^{\mathbf{R}}_i = c^{\mathbf{S}}_i \oplus s^{\mathbf{R}}_{\delta i} = \mathtt{sh}^{\mathbf{S}}_i$.

2. $w_i \in \mathcal{Y}$ *is corrupted and* $s^{\mathbf{R}}_{\delta i} \neq s^{\mathbf{S}}_{\delta i}$: we claim that in this case, $support((\mathtt{sh'}^{\mathbf{R}}_{\delta,i,1}, \ldots, \mathtt{sh'}^{\mathbf{R}}_{\delta,i,n}) - (\mathtt{sh}^{\mathbf{R}}_{\delta,i,1}, \ldots, \mathtt{sh}^{\mathbf{R}}_{\delta,i,n})) \notin \Gamma_{top}$ and hence $\mathtt{sh}^{\mathbf{R}}_i = \perp$. On the contrary, let

$$A_1 = support((\mathtt{sh'}^{\mathbf{R}}_{\delta,i,1}, \ldots, \mathtt{sh'}^{\mathbf{R}}_{\delta,i,n}) - (\mathtt{sh}^{\mathbf{R}}_{\delta,i,1}, \ldots, \mathtt{sh}^{\mathbf{R}}_{\delta,i,n})) \in \Gamma_{top}.$$

   Also suppose that the adversary actually corrupted $A_2 \in \Gamma_{top}$, where $w_i \in A_2$. Let $B = \mathcal{Y} \setminus (A_1 \cup A_2)$. We claim that $B \notin \Gamma$. Otherwise, we have three sets, namely $B \in \Gamma$ and $A_1, A_2 \in \Gamma_{top}$, such that $B \cup A_1 \cup A_2 = \mathcal{Y}$ and two different sets in Γ, namely $A_1 \cup \mathcal{X}$ and $A_2 \cup \mathcal{X}$, each of which covers the entire bottom band. But this violates second condition of **N&S-II**, which is a contradiction. However, we now get that two different secrets $s^{\mathbf{R}}_{\delta i} \neq s^{\mathbf{S}}_{\delta i}$ have same shares corresponding to the channels in B which constitutes an access set, which is a contradiction.

Thus using all $\mathtt{sh}^{\mathbf{R}}_i$s, such that $\mathtt{sh}^{\mathbf{R}}_i \neq \perp$, $\mathbf{R}$ will correctly reconstruct $m^{\mathbf{R}} = m^{\mathbf{S}}$. □

Lemma 4.7 (Perfect Security)**.** *The* 3*-round protocol in directed network provides perfect secrecy.*

Proof. The secrecy of $m^{\mathbf{S}}$ depends upon the secrecy of the the secrets of block $B^{\mathbf{S}}_\delta$ communicated over honest wires, since knowing the other secrets of $B^{\mathbf{S}}_\delta$ allows the adversary to compute shares of $m^{\mathbf{S}}$ corresponding to an adversary set and by the property of MSP these shares leaks no information about $m^{\mathbf{S}}$. Recall that δ is the minimum index from $\{1, \ldots, n^2u+1\}$ that does not figure in any of the non-discarded conflict list received by $\mathbf{S}$. As shown in Claim 4.3, such a δ exists.

For a secret of $B^{\mathbf{S}}_\delta$ sent over an honest channel, adversary will know the shares of the secret corresponding to the corrupted channels. However, from the properties of MSP, these shares will not provide any information about the secrets of honest channels. Now during **Round II** and **Round III**, no other information about the shares of the secrets in $B^{\mathbf{S}}_\delta$, are exchanged between $\mathbf{S}$ and $\mathbf{R}$. Thus adversary will have no information about the secrets in $B^{\mathbf{S}}_\delta$, corresponding to the honest channels, which constitutes an access set. Consequently, adversary will not know the shares of $m^{\mathbf{S}}$ corresponding to these honest channels (even though he knows all the masked shares as they are broadcasted). This implies perfect security for $m^{\mathbf{S}}$. □

Theorem 4.8. *Suppose* **N&S-II** *is satisfied. Then the above protocol is a* 3*-round PSMT protocol which sends a message containing* $\ell = 1$ *field element. The protocol is efficient and performs computation and communication which is polynomial in* $|\mathcal{Y}|, |\mathcal{X}|$ *and size of the MSP* $\mathcal{M}$*.*

5 Efficient Three Round PSMT in Undirected Network

We now briefly discuss the idea of our 3-round PSMT in undirected network, which can send a message containing $\ell > 1$ field elements concurrently. We describe the protocol in **APPENDIX B**. Since the network is undirected, we have N *bi-directional* channels between $\mathbf{S}$ and $\mathbf{R}$ and the adversary structure will satisfy $\mathcal{Q}^2$ condition. Thus both way reliable communication will be possible. Now the first round of the protocol will be same as in directed network, except that $\mathbf{S}$ now sends *only one* block of values $B^{\mathbf{S}}$. $\mathbf{R}$ will then construct the conflict list and now he can reliably send back the conflict list by broadcasting it during second round. This will be correctly received by $\mathbf{S}$ and hence there is no chance of guessing attack. From the conflict list, $\mathbf{S}$ will correctly come to know all the secrets which were correctly received by $\mathbf{R}$ during first round. Now out of these secrets, the secrets which were communicated over the honest channels will be unknown to the adversary. So $\mathbf{S}$ applies randomness extracting algorithm [23, 1] on the secrets which were correctly received by $\mathbf{R}$ to construct an information theoretically secure pad of size more than one. Using this pad, $\mathbf{S}$ can securely send a message containing more than one field element. See **APPENDIX B** for details.

6 Conclusion and Open Problems

We showed that *pseudo-basis*, a fundamental and useful concept in coding theory and PSMT, does not extend to general adversary settings. As a result, many of the protocols of [25] tolerating general adversary in both directed and undirected network are flawed. We repair the situation by presenting an efficient 3-round protocol in directed network tolerating guessing attack, while the existing protocol in the same scenario (without using pseudobasis) was

exponential in the size of adversary structure. We also present an efficient 3-round PSMT protocol for multiple message in undirected network. The existing 2-round protocol of [25] (without using pseudobasis) in the same settings is capable of sending only single message at a time. An important open problem is to achieve 2-round protocol in the above settings for multiple messages.

Another important open problem is to obtain a tight upper bound on the size of MSP, which is a long standing open problem. It would also be interesting to see whether one can further improve the communication complexity of our protocols using *generalized broadcast* [23, 1, 13]. Designing efficient PSMT protocols in asynchronous settings against general adversary is also an interesting problem.

References

[1] S. Agarwal, R. Cramer, and R. de Haan. Asymptotically optimal two round perfectly secure message transmission. In C. Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*, pages 394–408. Springer-Verlag, 2006.

[2] A. Choudhury. Protocols for reliable and secure message transmission. Cryptology ePrint Archive, Report 2010/281, 2010.

[3] R. Cramer, I. Damgård, and U. M. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In *EUROCRYPT*, volume 1807 of *Lecture Notes in Computer Science*, pages 316–334, 2000.

[4] Y. Desmedt, S. Erotokritou, and R. Safavi-Naini. Simple and communication complexity efficient almost secure and perfectly secure message transmission schemes. In D. J. Bernstein and T. Lange, editors, *Progress in Cryptology - AFRICACRYPT 2010, Third International Conference on Cryptology in Africa, Stellenbosch, South Africa, May 3-6, 2010. Proceedings*, volume 6055 of *Lecture Notes in Computer Science*, pages 166–183. Springer, 2010.

[5] Y. Desmedt and Y. Wang. Perfectly secure message transmission revisited. In E. Biham, editor, *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 502–517. Springer, 2003.

[6] Y. Desmedt, Y. Wang, and M. Burmester. A complete characterization of tolerable adversary structures for secure point-to-point transmissions without feedback. In *Algorithms and Computation, 16th International Symposium, ISAAC 2005, Sanya, Hainan, China, December 19-21, 2005, Proceedings*, volume 3827 of *Lecture Notes in Computer Science*, pages 277–287. Springer Verlag, 2005.

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

[8] M. Fitzi, M. K. Franklin, J. A. Garay, and S. H. Vardhan. Towards optimal and efficient perfectly secure message transmission. In S. P. Vadhan, editor, *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, volume 4392 of *Lecture Notes in Computer Science*, pages 311–322. Springer, 2007.

[9] 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.

[10] M. V. N. A. Kumar, P. R. Goundan, K. Srinathan, and C. Pandu Rangan. On perfectly secure communication over arbitrary networks. In *PODC 2002, Proceedings of the Twenty-First Annual ACM Symposium on Principles of Distributed Computing, July 21-24, 2002 Monterey, California, USA*, pages 193–202. ACM, 2002.

[11] K. Kurosawa. General error decodable secret sharing scheme and its application. Cryptology ePrint Archive, Report 2009/263, 2009.

[12] K. Kurosawa. Round-efficient perfectly secure message transmission scheme against general adversary. Cryptology ePrint Archive, Report 2010/450, 2010.

[13] K. Kurosawa and K. Suzuki. Truly efficient 2 round perfectly secure message transmission scheme. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 324–340. Springer, 2008.

[14] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error Correcting Codes*. North-Holland Publishing Company, 1978.

[15] A. Patra, A. Choudhary, and C. Pandu Rangan. Constant phase efficient protocols for secure message transmission in directed networks. In I. Gupta and R. Wattenhofer, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Principles of Distributed Computing, PODC 2007, Portland, Oregon, USA, August 12-15, 2007*, pages 322–323. ACM, 2007.

[16] A. Patra, A. Choudhary, and C. Pandu Rangan. Brief announcement: Perfectly secure message transmission in directed networks revisited. In S. Tirthapura and L. Alvisi, editors, *Proceedings of the 28th Annual ACM Symposium on Principles of Distributed Computing, PODC 2009, Calgary, Alberta, Canada, August 10-12, 2009*, pages 305–328. ACM, 2009.

[17] A. Patra, A. Choudhary, and C. Pandu Rangan. On communication complexity of secure message transmission in directed networks. In K. Kant, S. V. Premmaraju, K. M. Sivalingam, and J. Wu, editors, *Distributed Computing and Networking, 11th International Conference, ICDCN 2010, Kolkata, India, January 3 - 6, 2010, Proceedings*, volume 5935 of *Lecture Notes in Computer Science*, pages 42–53. Springer Verlag, 2010.

[18] A. Patra, A. Choudhary, K. Srinathan, and C. Pandu Rangan. Constant phase bit optimal protocols for perfectly reliable and secure message transmission. In R. Barua and T. Lange, editors, *Progress in Cryptology - INDOCRYPT 2006, 7th International Conference on Cryptology in India, Kolkata, India, December 11-13, 2006, Proceedings*, volume 4329 of *Lecture Notes in Computer Science*, pages 221–235. Springer, 2006.

[19] A. Patra, A. Choudhary, M. Vaidyanathan, and C. Pandu Rangan. Efficient perfectly reliable and secure message transmission tolerating mobile adversary. In Y. Mu, W. Susilo, and J. Seberry, editors, *Information Security and Privacy, 13th Australasian Conference, ACISP 2008, Wollongong, Australia, July 7-9, 2008, Proceedings*, volume 5107 of *Lecture Notes in Computer Science*, pages 170–186. Springer, 2008.

[20] A. Patra, B. Shankar, A. Choudhary, K. Srinathan, and C. Pandu Rangan. Perfectly secure message transmission in directed networks tolerating threshold and non threshold adversary. In F. Bao, S. Ling, T. Okamoto, H. Wang, and C. Xing, editors, *Cryptology and Network Security, 6th International Conference, CANS 2007, Singapore, December 8-10, 2007, Proceedings*, volume 4856 of *Lecture Notes in Computer Science*, pages 80–101. Springer, 2007.

[21] H. Sayeed and H. Abu-Amara. Efficient perfectly secure message transmission in synchronous networks. *Information and Computation*, 126(1):53–61, 1996.

[22] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[23] K. Srinathan, A. Narayanan, and C. Pandu Rangan. Optimal perfectly secure message transmission. In M. K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 545–561. Springer, 2004.

[24] Q. Yang and Y. Desmedt. Cryptanalysis of secure message transmission protocols with feedback. In *ICITS*, volume 5973 of *Lecture Notes in Computer Science*, pages 159–176, 2009.

[25] Q. Yang and Y. Desmedt. General perfectly secure message transmission using linear codes. In *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 448–465, 2010.

A APPENDIX A: Properties of MSP

Theorem 3.1: *Two different secrets shared according to an MSP realizing* Σ *in an* LSSS *cannot have same shares corresponding to an access set in* Σ.

PROOF: Let $\mathcal{M}$ be the $d \times e$ matrix realizing the access structure Σ and let s_1 and s_2 be two secrets, such that

$s_1 \neq s_2$. Without loss of generality, let $\{P_1, \ldots, P_t\}$ be an access set and let the first t rows of $\mathcal{M}$ be assigned to $\{P_1, \ldots, P_t\}$, with i^{th} row being assigned to party P_i, for $i = 1, \ldots, t$. Let $\mathcal{M}_{acc}$ denote the submatrix of $\mathcal{M}$, containing only the rows assigned to the parties in the access set (namely the first t rows). Let $(\mathtt{sh}_{1,1}, \ldots, \mathtt{sh}_{1,t})$ and $(\mathtt{sh}_{2,1}, \ldots, \mathtt{sh}_{2,t})$ be the shares of s_1 and s_2 respectively according to the MSP $\mathcal{M}$, corresponding to the access set $\{P_1, \ldots, P_t\}$. More specifically, let $v_1 = (s_1, \vec{\rho_1})$ and $v_2 = (s_2, \vec{\rho_2})$, where each $\vec{\rho_i} \in \mathbb{F}^{e-1}$ is a random vector. Then

$$(\mathtt{sh}_{1,1}, \ldots, \mathtt{sh}_{1,t})^T = \mathcal{M}_{acc} \cdot (v_1)^T. \tag{1}$$

and

$$(\mathtt{sh}_{2,1}, \ldots, \mathtt{sh}_{2,t})^T = \mathcal{M}_{acc} \cdot (v_2)^T. \tag{2}$$

According to the theorem, $(\mathtt{sh}_{1,1}, \ldots, \mathtt{sh}_{1,t})^T \neq (\mathtt{sh}_{2,1}, \ldots, \mathtt{sh}_{2,t})^T$. On the contrary, assume that $(\mathtt{sh}_{1,1}, \ldots, \mathtt{sh}_{1,t})^T = (\mathtt{sh}_{2,1}, \ldots, \mathtt{sh}_{2,t})^T$. This implies that

$$\begin{aligned} (\mathtt{sh}_{1,1}, \ldots, \mathtt{sh}_{1,t})^T - (\mathtt{sh}_{2,1}, \ldots, \mathtt{sh}_{2,t})^T &= (0, \ldots, 0(t \text{ times}))^T \\ &= \mathcal{M}_{acc} \cdot ((v_1)^T - (v_2)^T) \end{aligned}$$

Since $\{P_1, \ldots, P_t\}$ is an *access set*, there exists a *recombination vector*, say $\vec{\alpha}_{acc}$, such that $\vec{\alpha}_{acc} \cdot \mathcal{M}_{acc} = (1, 0, \ldots, 0)$. So multiplying both sides of the above equation by $\vec{\alpha}_{acc}$, we get

$$\vec{\alpha}_{acc} \cdot (0, \ldots, 0(t \text{ times}))^T = \vec{\alpha}_{acc} \cdot \mathcal{M}_{acc} \cdot ((v_1)^T - (v_2)^T)$$

The above equation further implies that

$$(0, \ldots, 0(t \text{ times}))^T = (1, 0, \ldots, 0) \cdot ((v_1)^T - (v_2)^T) = (s_1 - s_2, 0, \ldots, 0)^T$$

Thus, we get $s_1 - s_2 = 0$, which implies that $s_1 = s_2$, which is a contradiction. □

B APPENDIX B: Three Round PSMT in Undirected Network

Let **S** and **R** be connected by N bidirectional channels, denoted by $\mathcal{W} = \{w_1, \ldots, w_N\}$. Let $\mathcal{A}$ be a non-threshold adversary, specified by an adversary structure Γ over the set of channels $\mathcal{W}$. Moreover, let $\Sigma = \Gamma^c$ be the corresponding access structure over $\mathcal{W}$. Furthermore, let $\mathcal{A}$ satisfies $\mathcal{Q}^2$ condition with respect to $\mathcal{W}$. Let $(\mathcal{M}, \psi)$ be the MSP realizing the access structure Σ. *Without loss of generality and for simplicity, we assume that only i^{th} row of $\mathcal{M}$ is assigned to channel w_i, for $i = 1, \ldots, N$.* Thus,

$$\mathcal{M} = \begin{pmatrix} \vec{m_1} \\ \vdots \\ \vec{m_N} \end{pmatrix}$$

is an $N \times e$ matrix over $\mathbb{F}$. *However, our protocol can be easily generalized to the case when more than one row of $\mathcal{M}$ is assigned to some w_i.* Now we present a definition and a well known existing tool, which will be used in our protocol.

Definition B.1 (Size of $\mathcal{A}$ [25])**.** The size of $\mathcal{A}$, denoted by $sz^{\mathcal{A}}$, is defined as $sz^{\mathcal{A}} = \max\{\text{size of } B | B \in \Gamma\}$, where size of any $B \in \Gamma$ is defined as $|B|$. Thus, $sz^{\mathcal{A}}$ denotes the maximum number of channels that can be corrupted by the adversary. It is easy to see that $sz^{\mathcal{A}} = \mathcal{O}(N)$.

Definition B.2 (Extracting Randomness (EXTRAND) [23, 1])**.** Suppose **S** and **R** have some how agreed on a sequence of N elements $X = [X_1, \ldots, X_N]$, such that $N - F$ elements of X are known to the adversary and the remaining F elements are information theoretically secure, where $F < N$. However, neither **S** nor **R** knows which elements are information theoretically secure. Now both **S** and **R** wants to locally compute a sequence of F values, namely $Y = [Y_1, \ldots, Y_F]$, from X, such that the elements of Y are information theoretically secure. In [23], the authors presented an efficient algorithm, called $EXTRAND$ to address this. We use the following notation: $Y = EXTRAND_{N,F}(X)$.

B.1 3-Round Protocol for Multiple Message

We now present our three round efficient protocol called 3-Round-Multiple, which can *concurrently* send a message $m^{\mathbf{S}}$ containing $(N - sz^{\mathcal{A}})$ field elements. We first explain the idea of the protocol.

The Underlying Idea: During first round, $\mathbf{S}$ selects N random secrets and generates N shares for each secrets according to the LSSS $\mathcal{M}$. Now i^{th} secret along with the randomness used to compute the N shares for the ith secret, is sent over channel w_i. In addition, j^{th} share of each of the N secrets is sent over w_j. After receiving the values, $\mathbf{R}$ finds inconsistencies in the received information (in the same way as in our protocol for directed network). Then during second round, $\mathbf{R}$ broadcasts all inconsistencies to $\mathbf{S}$, which will be correctly received by $\mathbf{S}$. Now after comparing the inconsistent values (as received by $\mathbf{R}$), with the original values, $\mathbf{S}$ identifies all secrets which were incorrectly received by $\mathbf{R}$ during the first round. Now there are at least $(N - sz^{\mathcal{A}})$ information theoretically secure secrets among the secrets received by $\mathbf{R}$ correctly, since they are sent via honest channels. However, $\mathbf{S}$ does not know their exact identity. So $\mathbf{S}$ applies EXTRAND on the secrets to get a random and information theoretically secure one time pad consisting of $(N - sz^{\mathcal{A}})$ elements. Now using the pad, $\mathbf{S}$ masks elements of $m^{\mathbf{S}}$. Finally, during the third round, $\mathbf{S}$ broadcasts the masked message, along with the indices of the secrets which are used by $\mathbf{S}$ to generate the pad. Now using this information, $\mathbf{R}$ will also be able to correctly generate the pad and hence can get back $m^{\mathbf{S}}$ by unmasking the masked message with the pad.

A similar idea as above, is used in the three round protocol of [18] in threshold settings. So our protocol can be viewed as a generalization of the protocol of [18]. The protocol is given in Fig. 2.

We now prove the properties of the protocol.

Lemma B.3 (Security)**.** *In the protocol of Fig. 2, message $m^{\mathbf{S}}$ is information theoretically secure.*

Proof. To prove the lemma, we have to show that the pad $Y^{\mathbf{S}}$ is information theoretically secure. This can be proved by showing that adversary will have no information about $s_i^{\mathbf{S}}$'s which are sent over the channels not under his control. However this follows using similar arguments as used in the secrecy proof of our 3-round protocol in directed network. The proof now follows from the properties of EXTRAND algorithm and the fact that there are at least $N - sz^{\mathcal{A}}$ honest channels. □ □

Lemma B.4 (Reliability)**.** *In the protocol of Fig. 2, $\mathbf{R}$ will correctly recover $m^{\mathbf{R}} = m^{\mathbf{S}}$.*

Proof. To prove the lemma, we have to show that $X^{\mathbf{R}} = X^{\mathbf{S}}$. The rest will follow from the properties of EXTRAND. To show that $X^{\mathbf{R}} = X^{\mathbf{S}}$, we need to show that if w_i is a *corrupted* channel with $s_i^{\mathbf{R}} \neq s_i^{\mathbf{S}}$, then $w_i \in Faulty$. But this follows using similar arguments as used to prove part(2) of Claim 4.1. □

Theorem B.5. *The protocol in Fig. 2 is an efficient three round PSMT protocol, which sends a message containing $(N - sz^{\mathcal{A}})$ field elements by communicating $\mathcal{O}(N^3)$ field elements. The protocol performs computation which is polynomial in N and size of $\mathcal{M}$.*

Figure 2: Efficient Three Round Protocol Sending a Message of Size $(N - sz^{\mathcal{A}})$ Field Elements

Round I: S to R:

1. **S** selects N random elements $s_1^{\mathbf{S}}, \ldots, s_N^{\mathbf{S}}$ from $\mathbb{F}$ and computes $\mathsf{LSSS}(s_i^{\mathbf{S}}, \vec{\rho_i}^{\mathbf{S}}) = (\mathtt{sh}_{i1}^{\mathbf{S}}, \cdots, \mathtt{sh}_{in}^{\mathbf{S}})$, for $i = 1, \ldots, N$, where $\vec{\rho_i}^{\mathbf{S}}$'s are the randomness.
2. For $i = 1, \ldots, N$, **S** sends the following to **R** over channel w_i: (a) $(s_i^{\mathbf{S}}, \vec{\rho_i}^{\mathbf{S}})$ and (b) $\mathtt{sh}_{ji}^{\mathbf{S}}$, for $j = 1, \ldots, n$.

Round II: R to S:

1. For $i = 1, \ldots, N$, let **R** receive the following from **S** over channel w_i [a]: (a) $(s_i^{\mathbf{R}}, \vec{\rho_i}^{\mathbf{R}})$ and (b) $\mathtt{sh}_{ji}^{\mathbf{R}}$, for $j = 1, \ldots, n$.
2. For $i = 1, \ldots, N$, **R** computes $\mathsf{LSSS}(s_i^{\mathbf{R}}, \vec{\rho_i}^{\mathbf{R}}) = (\mathtt{sh}'^{\mathbf{R}}_{i1}, \cdots, \mathtt{sh}'^{\mathbf{R}}_{in})$ and checks $\mathtt{sh}'^{\mathbf{R}}_{ij} \stackrel{?}{=} \mathtt{sh}_{ij}^{\mathbf{R}}$, for $j = 1, \ldots, N$. If the test fails then **R** adds a conflicting tuple $(i, j, \mathtt{sh}'^{\mathbf{R}}_{ij}, \mathtt{sh}_{ij}^{\mathbf{R}})$ to a list $Conflict$, which is initially $\emptyset$.
3. Finally, **R** broadcasts the list $Conflict$, containing all conflicting values to **S**.

Round III: S to R:

1. **S** correctly receives the list $Conflict$. **S** initializes a list called $Faulty = \emptyset$.
2. For each $(i, j, \mathtt{sh}'^{\mathbf{R}}_{ij}, \mathtt{sh}_{ij}^{\mathbf{R}}) \in Conflict$, **S** does the following:
   (a) **S** checks $\mathtt{sh}'^{\mathbf{R}}_{ij} \stackrel{?}{=} \mathtt{sh}_{ij}^{\mathbf{S}}$. If the test fails then **S** concludes that **R** has received $s_i^{\mathbf{R}} \neq s_i^{\mathbf{S}}$ over w_i and adds w_i to list $Faulty$.
   (b) **S** checks $\mathtt{sh}_{ij}^{\mathbf{R}} \stackrel{?}{=} \mathtt{sh}_{ij}^{\mathbf{S}}$. If the test fails then **S** concludes that **R** has received $\mathtt{sh}_{ij}^{\mathbf{R}} \neq \mathtt{sh}_{ij}^{\mathbf{S}}$ over w_j and adds w_j to list $Faulty$.
3. **S** constructs a list $X^{\mathbf{S}}$, which is the collection of all $s_i^{\mathbf{S}}$'s, such that $w_i \notin Faulty$. **S** then computes $Y^{\mathbf{S}} = EXTRAND_{|X^{\mathbf{S}}|,(N-sz^{\mathcal{A}})}(X^{\mathbf{S}})$.
4. Let $Y^{\mathbf{S}} = (y_1^{\mathbf{S}}, \ldots, y_\ell^{\mathbf{S}})$ and the message $m^{\mathbf{S}} = (m_1^{\mathbf{S}}, \ldots, m_\ell^{\mathbf{S}})$, where $\ell = (N - sz^{\mathcal{A}})$. **S** computes $C^{\mathbf{S}} = (c_1^{\mathbf{S}}, \ldots, c_\ell^{\mathbf{S}}) = m^{\mathbf{S}} \oplus Y^{\mathbf{S}}$, where $c_l^{\mathbf{S}} = y_l^{\mathbf{S}} \oplus m_l^{\mathbf{S}}$, for $l = 1, \ldots, \ell$.
5. Finally **S** broadcasts $C^{\mathbf{S}}$ and the list $Faulty$ to **R** and terminates the protocol.

Message Recovery by R:

1. **R** correctly receives $C^{\mathbf{R}} = C^{\mathbf{S}}$ and the list $Faulty$.
2. **R** constructs a list $X^{\mathbf{R}}$, which is the collection of all $s_i^{\mathbf{R}}$'s, such that $w_i \notin Faulty$. **R** then computes $Y^{\mathbf{R}} = EXTRAND_{|X^{\mathbf{R}}|,(N-sz^{\mathcal{A}})}(X^{\mathbf{R}})$.
3. Finally **R** computes $m^{\mathbf{R}} = C^{\mathbf{R}} \oplus Y^{\mathbf{R}}$ and terminates the protocol.

[a] If channel w_i is not under the control of $\mathcal{A}$ then $(s_i^{\mathbf{R}}, \vec{\rho_i}^{\mathbf{R}}) = (s_i^{\mathbf{S}}, \vec{\rho_i}^{\mathbf{S}})$ and $\mathtt{sh}_{ji}^{\mathbf{R}} = \mathtt{sh}_{ji}^{\mathbf{S}}$, for $j = 1, \ldots, N$.