Multi-party PSM, Revisited: Improved Communication and Unbalanced Communication

Léonard Assouline, Tianren Liu

Multi-party PSM, Revisited: Improved Communication and Unbalanced Communication

Léonard Assouline
École Normale Supérieure, Paris
leonard.assouline@ens.fr

Tianren Liu
University of Washington, Seattle
tianrenl@uw.edu

October 5, 2021

Abstract

We improve the communication complexity in the Private Simultaneous Messages (PSM) model, which is a minimal model of non-interactive information-theoretic multi-party computation. The state-of-the-art PSM protocols were recently constructed by Beimel, Kushilevitz and Nissim (EUROCRYPT 2018).

We present new constructions of k-party PSM protocols. The new protocols match the previous upper bounds when $k = 2$ or 3 and improve the upper bounds for larger k. We also construct 2-party PSM protocols with unbalanced communication complexity. More concretely,

- For infinitely many k (including all $k \leq 20$), we construct k-party PSM protocols for arbitrary functionality $f : [N]^k \to \{0,1\}$, whose communication complexity is $O_k(N^{\frac{k-1}{2}})$. This improves the former best known upper bounds of $O_k(N^{\frac{k}{2}})$ for $k \geq 6$, $O(N^{7/3})$ for $k = 5$, and $O(N^{5/3})$ for $k = 4$.
- For all rational $0 < \eta < 1$ whose denominator is ≤ 20, we construct 2-party PSM protocols for arbitrary functionality $f : [N] \times [N] \to \{0,1\}$, whose communication complexity is $O(N^{\eta})$ for one party, $O(N^{1-\eta})$ for the other. Previously the only known unbalanced 2-party PSM has communication complexity $O(\log(N)), O(N)$.

1 Introduction

Private Simultaneous Messages (PSM) is a minimal model of secure multi-party computation. It was introduced by Feige, Kilian and Naor [FKN94], and was generalized to the multi-party setting by Ishai and Kushilevitz [IK97].

In a PSM protocol for evaluating a k-ary functionality $f : [N]^k \to \{0,1\}$, there are k parties. They all share a common random string. For all $i \in [k]$, the i-th party holds a private input x_i. There is additionally a special party, called the *referee*. The referee receives one message from each party and is able to compute $f(x_1, \ldots, x_k)$, and should learn no other information about $x_1, \ldots, x_k$.

PSM is studied as an information-theoretic primitive. The key complexity measure is the communication complexity. The common random string is crucial for the model as the common random string is the only mean to protect the privacy against an unbounded adversarial referee, when the k parties cannot communicate with each other.

In the PSM model, there are relatively efficient PSM protocols for computing non-deterministic branching programs [FKN94] and modular branching programs [IK97]. But for general functionalities, little is known regarding their communication complexity in the PSM model. Assuming every party holds an input in $[N]$, the best known lower bound of 2-party PSM is $3 \log N -$

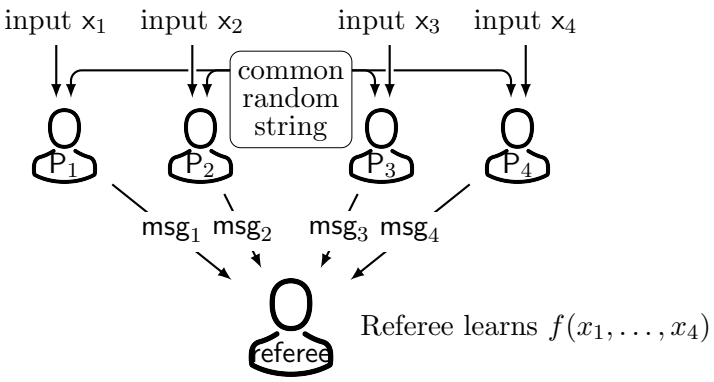

Figure 1: Illustration of a multi-party PSM protocol

$O(\log \log N)$ [AHMS20]. In k-party PSM where each party holds a 1-bit input, Ball et al. showed an $\Omega(k^2/\log k)$ lower bound [BHI+20]. Though the lower bounds are at most polynomial in the total input length, all known upper bounds are exponential, leaving an exponential gap between upper and lower bounds. For any functionality $f : [N]^k \to \{0,1\}$, a "naïve" k-party PSM requires $O(N^{k-1})$ communication (the 2-party version was presented in [FKN94]). The first novel upper bound is $O(\sqrt{N})$ for 2-party PSM [BIKK14], and it was recently generalized to an $O_k(N^{k/2})$ upper bound for k-party PSM [BKN18]. In the same paper, Beimel, Kushilevitz and Nissim also further optimize the communication complexity for small $k = 3, 4, 5$. In particular, they obtain an $O(N)$ upper bound for 3-party PSM. For $k = 4$ or 5, they improve the protocol by letting parties jointly emulate their 3-party PSM. Their results are summarized in Table 1.

1.1 Our Contributions

In the paper, we present two classes of results: We present new k-party PSM protocols that improve the communication complexity for infinitely many k. We introduce the notion of *unbalanced* 2-party PSM protocols, which allows a flexible repartition of the communication complexity among the two parties, and we such protocols.

k-party PSM protocols. We present a framework for constructing multi-party PSM. The new framework improves the communication complexity upper bounds for infinitely many k. To compute any k-ary functionality $f : [N]^k \to \{0,1\}$,

- For all $k \leq 20$, our framework yields a k-party PSM protocol of communication complexity $O(N^{\frac{k-1}{2}})$.
- For all k such that $k+1$ is a prime or a prime power, our framework yields a k-party PSM protocol of communication complexity $O_k(N^{\frac{k-1}{2}})$.
- For all k, we *conjecture* that our framework will yield a k-party PSM protocol of communication complexity $O_k(N^{\frac{k-1}{2}})$.

2-party unbalanced PSM protocols. We also present a framework for constructing 2-party PSM protocols with unbalanced communication complexity. The new framework allows us to reduce the message length of one party at the cost of increasing the communication of the other party.

Number of parties	[BIKK14]	[BKN18]	This work
2	$O(N^{1/2})$	$O(N^{1/2})$	$O(N^{1/2})$
3		$O(N)$	$O(N)$
4		$O(N^{5/3})$	$O(N^{3/2})$
5		$O(N^{7/3})$	$O(N^2)$
$k \geq 6$		$O(\text{poly}(k) \cdot N^{k/2})$	$2^{O(k)} \cdot N^{\frac{k-1}{2}}$ for infinitely many k including all $k \leq 20$

Table 1: The communication complexity of computing general $f : [N]^k \to \{0,1\}$ in multi-party PSM model

	Communication complexity of one party	Communication complexity of the other party
[FKN94]	$O(\log N)$	N
[BIKK14]	$O(N^{1/2})$	$O(N^{1/2})$
This work	$O(N^{\eta})$	$O(N^{1-\eta})$

Table 2: The unbalanced communication complexity of general $f : [N] \times [N] \to \{0,1\}$ in 2-party PSM model

We offer an almost smooth trade-off between the communication complexity of the two parties. To compute any functionality $f : [N] \times [N] \to \{0,1\}$,

- For every rational $\eta \in (0,1)$ whose denominator is no more than 20, our framework yields a 2-party PSM protocol, where one party sends $O(N^{\eta})$ bits and the other sends $O(N^{1-\eta})$ bits.
- For every rational $\eta \in (0,1)$, we *conjecture* that our framework will yield a 2-party PSM protocol, where one party sends $O_{\eta}(N^{\eta})$ bits and the other sends $O_{\eta}(N^{1-\eta})$ bits.

To some extent, such a trade-off was known in the literature when $\eta = 0$. The first 2-party PSM protocol is of communication complexity $O(N)$ but is strongly unbalanced: one of the two parties only sends $O(\log N)$ bits [FKN94].

1.2 Proof Overview

This section presents the main ideas behind our new multi-party PSM protocols. We start with a warm-up example of a 3-party PSM, which is originally constructed by [BKN18]. We present it in a way that matches the framework we will later introduce. Then we present a new 5-party PSM to demonstrate the power of our framework. The 5-party PSM example relies on new technique such as "hard terms cancelling". It can be easily generalized into a framework for constructing k-party PSM protocols for any odd k. But we do not formally present this framework in the paper.

Instead, in section 3, we develop a modified framework that supports odd as well as even values of k. The modified framework evenly divides every party's input into two halves, this idea was first introduced in [BIKK14]. When we formally present the modified framework in Section 3.1, we use a 4-party PSM as an example.

In section 4, we develop another framework for constructing unbalanced 2-party PSM protocols. Most terminologies and techniques are shared between the framework for k-party and the framework for unbalanced 2-party. Informally, the unbalanced 2-party PSM framework is the "tensor product"

of two copies of the k-party framework. When we present the new framework in Section 4.1, we use as an exmaple a 2-party PSM with unbalanced communication $O(N^{1/3}), O(N^{2/3})$.

Background: 3-Party PSM [BKN18]. In this 3-party PSM protocol, three parties hold $x_1, x_2, x_3 \in [N]$ respectively. The protocol takes $O(N)$ communication and allows the referee to learn $f(x_1, x_2, x_3)$.

Fix a finite field $\mathbb{F}$. Let the i-th party locally computes a unit vector $\mathbf{x}_i \in \mathbb{F}^N$. That is, all entries in $\mathbf{x}_i$ are zero except for $\mathbf{x}_i[x_i] = 1$. Let $\mathbf{F}$ be the truth table of f represented as an $N \times N \times N$ array, we have $f(x_1, x_2, x_3) = \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \rangle$, where $\otimes$ denotes the tensor product and $\langle \cdot, \cdot \rangle$ denotes the inner product.

Therefore, it is sufficient to construct a 3-party PSM protocol, where the i-th party has input $\mathbf{x}_i \in \mathbb{F}^N$ (not necessarily being an unit vector) and the referee learns $\langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \rangle$ for some public $\mathbf{F} \in \mathbb{F}^{N \times N \times N}$.

We start by letting the i-th party sample random $\mathbf{r}_i \in \mathbb{F}^N$ and send the one-time padded $\bar{\mathbf{x}}_i := \mathbf{x}_i + \mathbf{r}_i$ to the referee. Then the referee can compute $\langle \mathbf{F}, \bar{\mathbf{x}}_1 \otimes \bar{\mathbf{x}}_2 \otimes \bar{\mathbf{x}}_3 \rangle$. We call this term a "*masked term*", because it is computed from the masked inputs $\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \bar{\mathbf{x}}_3$. This masked term can be decomposed as the sum of several "*pure terms*"

$$\begin{aligned}\langle \mathbf{F}, \bar{\mathbf{x}}_1 \otimes \bar{\mathbf{x}}_2 \otimes \bar{\mathbf{x}}_3 \rangle = \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \rangle + \\ \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{r}_3 \rangle + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{r}_2 \otimes \mathbf{x}_3 \rangle + \langle \mathbf{F}, \mathbf{r}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \rangle + \\ \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{r}_2 \otimes \mathbf{r}_3 \rangle + \langle \mathbf{F}, \mathbf{r}_1 \otimes \mathbf{x}_2 \otimes \mathbf{r}_3 \rangle + \langle \mathbf{F}, \mathbf{r}_1 \otimes \mathbf{r}_2 \otimes \mathbf{x}_3 \rangle + \\ \langle \mathbf{F}, \mathbf{r}_1 \otimes \mathbf{r}_2 \otimes \mathbf{r}_3 \rangle.\end{aligned} \tag{1}$$

We classify the pure terms into two categories:

Target Term The term $\langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \rangle$. It is the term that the referee should learn as a consequence of the 3-party PSM protocol.

Easy Term All the other terms fall into this category. As the name suggested, there also exist "hard terms", which will be introduced in the next example of 5-party PSM.

The easy terms are called "easy" because each of them can be securely revealed to the referee using only $O(N)$ communication. More formally, let the parties additionally sample random $r_1, \ldots, r_7 \in \mathbb{F}$ from their common random string such that $r_1 + \cdots + r_7 = 0$. There exist sub-protocols revealing each of

$$\begin{gathered}r_1 + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{r}_3 \rangle,\ r_2 + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{r}_2 \otimes \mathbf{x}_3 \rangle,\ r_3 + \langle \mathbf{F}, \mathbf{r}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \rangle, \\ r_4 + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{r}_2 \otimes \mathbf{r}_3 \rangle,\ r_5 + \langle \mathbf{F}, \mathbf{r}_1 \otimes \mathbf{x}_2 \otimes \mathbf{r}_3 \rangle,\ r_6 + \langle \mathbf{F}, \mathbf{r}_1 \otimes \mathbf{r}_2 \otimes \mathbf{x}_3 \rangle, \\ r_7 + \langle \mathbf{F}, \mathbf{r}_1 \otimes \mathbf{r}_2 \otimes \mathbf{r}_3 \rangle\end{gathered} \tag{2}$$

to the referee without leaking any other information, taking at most $O(N)$ communication.

Assume that such sub-protocols exist, we can easily finish the 3-party PSM: The i-th party sends $\bar{\mathbf{x}}_i := \mathbf{x}_i + \mathbf{r}_i$, they use the the aforementioned sub-protocols to reveal (2). The correctness follows almost directly from (1).

The only missing piece is to contruct sub-protocols for computing the terms in (2). Let us discuss them individually:

- For the last term $r_7 + \langle \mathbf{F}, \mathbf{r}_1 \otimes \mathbf{r}_2 \otimes \mathbf{r}_3 \rangle$, any party (e.g. the first party) can compute it and send it to the referee.

- For the term $r_4 + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{r}_2 \otimes \mathbf{r}_3 \rangle$, the first party computes it and sends it to the referee. Similarly for $r_5 + \langle \mathbf{F}, \mathbf{r}_1 \otimes \mathbf{x}_2 \otimes \mathbf{r}_3 \rangle$ and $r_6 + \langle \mathbf{F}, \mathbf{r}_1 \otimes \mathbf{r}_2 \otimes \mathbf{x}_3 \rangle$.
- For the term $r_1 + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{r}_3 \rangle$, both first and second party need to participate. Since the first party knows $\mathbf{F}, \mathbf{x}_1, \mathbf{r}_3$, it can locally compute a vector $\mathbf{g} \in \mathbb{F}^N$ such that

$$r_1 + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{r}_3 \rangle = r_1 + \langle \mathbf{g}, \mathbf{x}_2 \rangle.$$

  Then they can jointly reveal it to the referee using the PSM for inner product (more details are provided in Section B.1). Similarly for $r_2 + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{r}_2 \otimes \mathbf{x}_3 \rangle$ and $r_3 + \langle \mathbf{F}, \mathbf{r}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \rangle$.

Example: 5-Party PSM. We will sketch a 5-party PSM protocol for any $f : [N]^5 \to \{0, 1\}$ with communication complexity $O(N^2)$.

Let $\mathbb{F}$ be a finite field. Following the same observation we made in the 3-party PSM example, it is sufficient to construct a PSM protocol for any function of the form $(\mathbf{x}_1, \ldots, \mathbf{x}_5) \mapsto \langle \mathbf{F}, \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_5 \rangle$, where $\otimes$ denotes the *tensor product*, the i-th party having input $\mathbf{x}_i \in \mathbb{F}^N$, $\mathbf{F}$ is public and fixed being the truth table of f.

For each $\Omega \subseteq \{1, 2, 3, 4, 5\}$, parties sample a dimension-$|\Omega|$ tensor $\mathbf{R}_\Omega \in \mathbb{F}^{N^{|\Omega|}}$ from the common random string. Define $\bar{\mathbf{X}}_\Omega := \mathbf{R}_\Omega + \bigotimes_{i \in \Omega} \mathbf{x}_i$. For example, $\bar{\mathbf{X}}_{\{2\}} := \mathbf{R}_{\{2\}} + \mathbf{x}_2$ and $\bar{\mathbf{X}}_{\{3,4\}} := \mathbf{R}_{\{3,4\}} + \mathbf{x}_3 \otimes \mathbf{x}_4$. Since the communication budget is $O(N^2)$, they can perform a PSM sub-protocol so that the referee learns $\bar{\mathbf{X}}_\Omega$ for all Ω such that $|\Omega| \leq 2$.

Learning those tensors allows the referee to compute many terms, including $\langle \mathbf{F}, \bar{\mathbf{X}}_{\{1,2\}} \otimes \bar{\mathbf{X}}_{\{3,4\}} \otimes \bar{\mathbf{X}}_{\{5\}} \rangle$. This term can be decomposed into the sum of the following 8 terms:

$$\begin{aligned}
&\langle \mathbf{F}, \bar{\mathbf{X}}_{\{1,2\}} \otimes \bar{\mathbf{X}}_{\{3,4\}} \otimes \bar{\mathbf{X}}_{\{5\}} \rangle \\
&= \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{x}_4 \otimes \mathbf{x}_5 \rangle + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{x}_4 \otimes \mathbf{R}_{\{5\}} \rangle \\
&\quad + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{R}_{\{3,4\}} \otimes \mathbf{x}_5 \rangle + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{R}_{\{3,4\}} \otimes \mathbf{R}_{\{5\}} \rangle \\
&\quad + \langle \mathbf{F}, \mathbf{R}_{\{1,2\}} \otimes \mathbf{x}_3 \otimes \mathbf{x}_4 \otimes \mathbf{x}_5 \rangle + \langle \mathbf{F}, \mathbf{R}_{\{1,2\}} \otimes \mathbf{x}_3 \otimes \mathbf{x}_4 \otimes \mathbf{R}_{\{5\}} \rangle \\
&\quad + \langle \mathbf{F}, \mathbf{R}_{\{1,2\}} \otimes \mathbf{R}_{\{3,4\}} \otimes \mathbf{x}_5 \rangle + \langle \mathbf{F}, \mathbf{R}_{\{1,2\}} \otimes \mathbf{R}_{\{3,4\}} \otimes \mathbf{R}_{\{5\}} \rangle.
\end{aligned} \tag{3}$$

Any term that is formed in the same way as the left-hand side of (3), i.e. $\langle \mathbf{F}, \bar{\mathbf{X}}_{S_1} \otimes \cdots \otimes \bar{\mathbf{X}}_{S_t} \rangle$ for some $S_1 + \cdots + S_t = \{1, 2, 3, 4, 5\}$, is called a *masked term*. It can be computed by the referee if $|S_i| \leq 2$ for all i.

Any term that is formed in the same way as the right-hand side of (3), i.e. $\langle \mathbf{F}, \mathbf{R}_{S_1} \otimes \cdots \otimes \mathbf{R}_{S_t} \otimes \mathbf{x}_{i_1} \otimes \cdots \otimes \mathbf{x}_{i_w} \rangle$ for some $S_1 + \cdots + S_t + \{i_1, \ldots, i_w\} = \{1, 2, 3, 4, 5\}$, is called a *pure term*. As hinted by equation (3), every masked term is equal to the sum of 2^t pure terms.

The pure terms fall naturally into three categories. In particular, we introduce a new category called *hard terms*.

Target term The term $\langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{x}_4 \otimes \mathbf{x}_5 \rangle$ is called the target term.

Easy term A pure term $\langle \mathbf{F}, \mathbf{R}_{S_1} \otimes \cdots \otimes \mathbf{R}_{S_t} \otimes \mathbf{x}_{i_1} \otimes \cdots \otimes \mathbf{x}_{i_w} \rangle$ is easy if $w \leq 3$. Every easy term can be computed using a PSM protocol with communication complexity $O(N^2)$. For example, $\langle \mathbf{F}, \mathbf{R}_{\{1,2\}} \otimes \mathbf{x}_3 \otimes \mathbf{x}_4 \otimes \mathbf{x}_5 \rangle$ is an easy term. The 5^{th} party, based on its view, can compute a tensor $\mathbf{G} \in \mathbb{F}^{N^2}$ such that $\langle \mathbf{F}, \mathbf{R}_{\{1,2\}} \otimes \mathbf{x}_3 \otimes \mathbf{x}_4 \otimes \mathbf{x}_5 \rangle = \langle \mathbf{G}, \mathbf{x}_3 \otimes \mathbf{x}_4 \rangle$. And $\langle \mathbf{G}, \mathbf{x}_3 \otimes \mathbf{x}_4 \rangle$ can be computed using a PSM protocol (Section B.1) with communication complexity $O(N^2)$.

Hard term Any pure term that is neither the target term nor an easy term.

Let us ignore the easy terms for now. Then equation (3) can be rewritten as

$$\begin{aligned}&\langle \mathbf{F}, \bar{\mathbf{X}}_{\{1,2\}} \otimes \bar{\mathbf{X}}_{\{3,4\}} \otimes \bar{\mathbf{X}}_{\{5\}} \rangle \\ &= \langle \mathbf{F}, \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_5 \rangle + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{x}_4 \otimes \mathbf{R}_{\{5\}} \rangle + \text{easy terms}.\end{aligned}$$

There is only one hard term left. We would like to cancel out the hard term by combining a few masked terms. Let us consider the following masked terms: $\langle \mathbf{F}, \bar{\mathbf{X}}_{\{1,2\}} \otimes \bar{\mathbf{X}}_{\{3\}} \otimes \bar{\mathbf{X}}_{\{4,5\}} \rangle$, $\langle \mathbf{F}, \bar{\mathbf{X}}_{\{1,2\}} \otimes \bar{\mathbf{X}}_{\{3,5\}} \otimes \bar{\mathbf{X}}_{\{4\}} \rangle$ and $\langle \mathbf{F}, \bar{\mathbf{X}}_{\{1,2\}} \otimes \bar{\mathbf{X}}_{\{3\}} \otimes \bar{\mathbf{X}}_{\{4\}} \otimes \bar{\mathbf{X}}_{\{5\}} \rangle$.

$$\begin{aligned}&\langle \mathbf{F}, \bar{\mathbf{X}}_{\{1,2\}} \otimes \bar{\mathbf{X}}_{\{3\}} \otimes \bar{\mathbf{X}}_{\{4,5\}} \rangle \\ &\quad = \langle \mathbf{F}, \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_5 \rangle + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{R}_{\{3\}} \otimes \mathbf{x}_4 \otimes \mathbf{x}_5 \rangle + \text{easy terms}, \\ &\langle \mathbf{F}, \bar{\mathbf{X}}_{\{1,2\}} \otimes \bar{\mathbf{X}}_{\{3,5\}} \otimes \bar{\mathbf{X}}_{\{4\}} \rangle \\ &\quad = \langle \mathbf{F}, \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_5 \rangle + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{R}_{\{4\}} \otimes \mathbf{x}_5 \rangle + \text{easy terms}, \\ &\langle \mathbf{F}, \bar{\mathbf{X}}_{\{1,2\}} \otimes \bar{\mathbf{X}}_{\{3\}} \otimes \bar{\mathbf{X}}_{\{4\}} \otimes \bar{\mathbf{X}}_{\{5\}} \rangle \\ &\quad = \langle \mathbf{F}, \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_5 \rangle + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{R}_{\{3\}} \otimes \mathbf{x}_4 \otimes \mathbf{x}_5 \rangle \\ &\qquad + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{R}_{\{4\}} \otimes \mathbf{x}_5 \rangle + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{x}_4 \otimes \mathbf{R}_{\{5\}} \rangle \\ &\qquad + \text{easy terms}.\end{aligned}$$

By carefully combining these masked tensors, we have

$$\begin{aligned}&\langle \mathbf{F}, \bar{\mathbf{X}}_{\{1,2\}} \otimes \bar{\mathbf{X}}_{\{3,4\}} \otimes \bar{\mathbf{X}}_{\{5\}} \rangle + \langle \mathbf{F}, \bar{\mathbf{X}}_{\{1,2\}} \otimes \bar{\mathbf{X}}_{\{3\}} \otimes \bar{\mathbf{X}}_{\{4,5\}} \rangle \\ &+ \langle \mathbf{F}, \bar{\mathbf{X}}_{\{1,2\}} \otimes \bar{\mathbf{X}}_{\{3,5\}} \otimes \bar{\mathbf{X}}_{\{4\}} \rangle - \langle \mathbf{F}, \bar{\mathbf{X}}_{\{1,2\}} \otimes \bar{\mathbf{X}}_{\{3\}} \otimes \bar{\mathbf{X}}_{\{4\}} \otimes \bar{\mathbf{X}}_{\{5\}} \rangle \\ &\qquad = 2 \cdot \langle \mathbf{F}, \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_5 \rangle + \text{easy terms}.\end{aligned} \tag{4}$$

Equation (4) shows us how to construct the desired PSM protocol. All of the masked tensors on the left-hand side of (4) can be computed by the referee. The parties perform a PSM sub-protocol so that the referee learns the sum of these easy terms. (The details are demonstrated in the last example of 3-party PSM, and are explained in Section 3.2.) Then from equation (4), the referee learns $2 \cdot \langle \mathbf{F}, \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_5 \rangle$.

As long as $\mathbb{F}$ is a finite field in which $2 \neq 0$, the referee has learned the target term. The protocol takes a communication cost of $O(N^2)$ field elements. □

1.3 Related Works

Besides [BIKK14, BKN18], our construction of PSM protocols is also inspired by the progress in Conditional Disclosure of Secrets (CDS). Until recently, CDS had a similar exponential gap between known upper and lower bounds. CDS can be viewed as a variant of PSM where the referee knows all but 1 bit of the input: Consider the 2-party case and let $[N]$ be the input domain for both parties. The upper bounds of $O(\sqrt{N})$ is conserved [BIKK14, GKW15]. A similar lower bound of $\Omega(\log N)$ is known [GKW15, AARV17]. Recently, Liu, Vaikuntanathan and Wee improved the CDS upper bound for arbitrary function to $2^{\tilde{O}\sqrt{\log N}}$ [LVW17]. In a slightly different setting, the amortized CDS upper bound per party is improved to $\Theta(1)$ [AARV17, AA18].

Gay, Kerenidis and Wee constructed 2-party CDS with smooth communication complexity trade-off between the two party [GKW15]. In particular, for any $\eta \in [0, 1]$, they constructed a 2-party CDS protocol where one party sends $O(N^\eta)$ bits and the other sends $O(N^{1-\eta})$ bits.

In [ABF+19, CGO21], constructions of *ad hoc* PSM are presented. In this framework, there are k parties, but only a subset of them will perform the computation. This notion, expanded in [BIK17], was shown to imply obfuscation.

2 Preliminaries

Let $\mathbb{N} := \{0, 1, \ldots\}$ denote the set of all natural numbers, and let $[n] := \{1, \ldots, n\}$. In this paper, $\mathbb{F}$ denotes a field, $\mathcal{R}$ denotes a finite commutative ring. For some prime power p, let $\mathbb{F}_p$ denote the unique finite field of order p. A vector will be denoted by a bold face lowercase letter. For a vector $\mathbf{v}$, let $\mathbf{v}[i]$ denote its i-th entry.

2.1 Tensor

A *tensor* refers to the generalization of vectors and matrices which have multiple indices. Roughly speaking, a tensor is a multi-dimensional array. In the paper, a tensor will be denoted by a bold face capital letter. A k-dimensional tensor $\mathbf{T} \in \mathbb{F}^{n_1 \times n_2 \times \ldots \times n_k}$ is essentially an array of size $n_1 \times n_2 \times \cdots \times n_k$. The entries in $\mathbf{T}$ are indexed by $(i_1, \ldots, i_k) \in [n_1] \times \cdots \times [n_k]$, and denoted by $\mathbf{T}[i_1, \ldots, i_k]$. A tensor can also be viewed as a representation of a multi-linear function: any k-linear function $f : \mathbb{F}^{n_1} \times \mathbb{F}^{n_2} \times \ldots \times \mathbb{F}^{n_k} \to \mathbb{F}$ can be uniquely determined by its coefficient tensor $\mathbf{F} \in \mathbb{F}^{n_1 \times \ldots \times n_k}$, such that

$$f(\mathbf{v}_1, \ldots, \mathbf{v}_k) = \sum_{i_1 \in [n_1], \cdots, i_k \in [n_k]} \mathbf{F}[i_1, \ldots, i_k] \cdot \mathbf{v}_1[i_1] \cdot \ldots \cdot \mathbf{v}_k[i_k]. \tag{5}$$

The inner product of two tensors $\mathbf{S}, \mathbf{T} \in \mathbb{F}^{n_1 \times n_2 \times \ldots \times n_k}$ is defined as

$$\langle \mathbf{S}, \mathbf{T} \rangle := \sum_{i_1 \in [n_1], \cdots, i_k \in [n_k]} \mathbf{S}[i_1, \ldots, i_k] \cdot \mathbf{T}[i_1, \ldots, i_k].$$

Given two tensors $\mathbf{S} \in \mathbb{F}^{n_1 \times \ldots \times n_k}$ and $\mathbf{T} \in \mathbb{F}^{m_1 \times \ldots \times m_\ell}$, we define $\mathbf{S} \otimes \mathbf{T}$, their tensor product. It is a tensor in $\mathbb{F}^{n_1 \times \ldots \times n_k \times m_1 \times \ldots \times m_\ell}$ such that

$$(\mathbf{S} \otimes \mathbf{T})[i_1, \ldots, i_k, j_1, \ldots, j_\ell] = \mathbf{S}[i_1, \ldots, i_k] \cdot \mathbf{T}[j_1, \ldots, j_\ell].$$

Using the notation of inner product and tensor product, Equation (5) can also be written as $f(\mathbf{v}_1, \ldots, \mathbf{v}_k) = \langle \mathbf{F}, \mathbf{v}_1 \otimes \ldots \otimes \mathbf{v}_k \rangle$.

2.2 Private Simultaneous Messages

Definition 1 (private simultaneous message)**.** A k-party functionality is a mapping $f : \mathcal{X}_1 \times \ldots \times \mathcal{X}_k \to \mathcal{Y}$, where $\mathcal{X}_1, \ldots, \mathcal{X}_k$ are its input spaces and $\mathcal{Y}$ is its output space.

A private simultaneous message (PSM) protocol for a functionality f consists of a randomness space $\mathcal{W}$ and a tuple of deterministic functions $(\mathsf{M}_1, \ldots, \mathsf{M}_k, \mathsf{R})$

$$\mathsf{M}_i : \mathcal{X}_i \times \mathcal{W} \to \{0,1\}^{\mathsf{cc}_i}, \quad \text{for all } i \in [k],$$
$$\mathsf{R} : \{0,1\}^{\mathsf{cc}_1} \times \ldots \{0,1\}^{\mathsf{cc}_k} \to \{0,1\},$$

where cc_i is the communication complexity of the i-th party, $\mathsf{cc} := \mathsf{cc}_1 + \ldots + \mathsf{cc}_k$ is the total communication complexity.

A perfectly secure PSM protocol for f satisfies the following properties:

(correctness.) For all input tuple $(x_1, \ldots, x_k) \in \mathcal{X}_1 \times \ldots \times \mathcal{X}_k$ and randomness $w \in \mathcal{W}$,

$$\mathsf{R}(\mathsf{M}_1(x_1, w), \ldots, \mathsf{M}_k(x_k, w)) = f(x_1, \ldots, x_k)$$

(privacy.) There exists a randomized simulator S, such that for any input $(x_1, \ldots, x_k) \in \mathcal{X}_1 \times \ldots \times \mathcal{X}_k$, the joint distribution of $\mathsf{M}_1(x_1, w), \ldots, \mathsf{M}_k(x_k, w)$ is the same as the distributions of $\mathsf{S}(f(x_1, \ldots, x_k))$, where the distributions are taken over $w \leftarrow \mathcal{W}$ and the coin tosses of S.

2.3 Randomized Encoding

Randomized encoding is a primitive closely relate to PSM. The randomized encoding of a function f is a randomized function $\hat{f}$. The output $\hat{f}(x, w)$, where w denotes the randomness, contains sufficient information to recover $f(x)$ and no other information about x.

Definition 2 (randomized encoding)**.** A randomized encoding for a function $f : \mathcal{X} \to \mathcal{Y}$ consists of a randomized encoding function $\hat{f} : \mathcal{X} \times \mathcal{W} \to \hat{\mathcal{Y}}$ and a deterministic decoding function $\mathsf{R} : \hat{\mathcal{Y}} \to \mathcal{Y}$, where $\mathcal{W}$ denotes the randomness space and $\hat{\mathcal{Y}}$ denotes the coding space.

A perfectly secure randomized encoding satisfies the following properties:

(correctness.) For all $x \in \mathcal{X}$ and randomness $w \in \mathcal{W}$,

$$\mathsf{R}(\hat{f}(x, w)) = f(x)$$

(privacy.) There exists a randomized simulator S, such that for any input $x \in \mathcal{X}$, the joint distribution of $\hat{f}(x, w)$ is the same as the distributions of $\mathsf{S}(f(x))$, where the distributions are taken over $w \leftarrow \mathcal{W}$ and the coin tosses of S.

Follows directly from the definitions, $(\mathsf{M}_1, \ldots, \mathsf{M}_k, \mathsf{R})$ is a PSM protocol for f if and only if $(\hat{f}, \mathsf{R})$ is a randomized encoding for f, where $\hat{f}(x_1, \ldots, x_k, w) := (\mathsf{M}_1(x_1, w), \ldots, \mathsf{M}_k(x_k, w))$.

In other words, PSM is a special form of randomized encoding, where the input is divided into a few portions, and each bit of the encoding only depends on the randomness and one portion of the input.

3 New Multi-party PSM Protocols

In this section, we present one of our main results: for many k, every functionality $f : [N]^k \to \{0, 1\}$ admits a PSM protocol of communication complexity $O_k(N^{\frac{k-1}{2}})$.

Theorem 3.1. *Let $f : [N]^k \to \{0, 1\}$ be an arbitrary k-party functionality.*

- *There is a k-party PSM protocol for f with communication and randomness complexity $O(N^{\frac{k-1}{2}})$, if $k \leq 20$.*
- *There is a k-party PSM protocol for f with communication and randomness complexity $O_k(N^{\frac{k-1}{2}})$, if $k + 1$ is a prime or a prime power.*

In this section, we prove a stronger statement. Let $\mathbb{F}$ be a finite field, consider the following auxiliary k-party functionality Aux_N^k:

k-party functionality Aux_N^k
• The i-th party has input $\mathbf{x}_{2i-1}, \mathbf{x}_{2i} \in \mathbb{F}^{\sqrt{N}}$
• The output is $\langle \mathbf{F}, \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_{2k} \rangle$, where $\mathbf{F}$ is public and fixed

As shown in the beginning of Section 3.1, a PSM protocol for Aux_N^k implies a PSM for $f : [N]^k \to \{0, 1\}$ with the same communication complexity. The reduction consists of having $\mathbf{F}$ be the truth table of f.

We will present a framework of constructing k-party PSM for Aux_N^k, whose communication complexity is $O_k(N^{\frac{k-1}{2}})$. Roughly speaking, the framework reduces the problem to a system of linear equations. A solution of the system implies a PSM protocol with the desired communication complexity. Therefore, we should rule out the possibility that the induced system has no solution. We partially achieve such a goal. We solve the induced system for infinitely many k:

- For all $k \leq 20$, we checked that the induced system of linear equations is solvable. For small k we solve the system by hand, and for larger k we verified it with a computer program.
- For all k such that $k+1$ is a prime power, we prove that the system is solvable.

Backed by the above observations, we strongly believe the induced system is solvable for all k.

Conjecture 1. Let $f : [N]^k \to \{0,1\}$ be an arbitrary k-party functionality. There is a k-party PSM protocol for f with communication and randomness complexity $O_k(N^{\frac{k-1}{2}})$.

Organization Section 3.1 presents our framework for constructing multi-party PSM, introduces new notations, and gives a 4-party PSM as a concrete example. The following Sections 3.2, 3.3, 3.4 are independent. Section 3.2 provides more technical detail of the PSM protocols yielded by our framework. Section 3.3 shows how the framework works for small k, and Section 3.4 shows how the framework works for any integer k such that $k+1$ is a prime power.

3.1 A Framework for Multi-party PSM

As mentioned in the beginning of Section 3, the functionality $f : [N]^k \to \{0,1\}$ can be reduced to functionality Aux_N^k. The reduction works as follows: Let $x_1, \ldots, x_k$ be the input, the j-th party has input $x_j \in [N]$. We evenly divide x_j into $x'_{2j-1}, x'_{2j} \in [\sqrt{N}]$. For each $i \in [2k]$, let $\mathbf{x}_i := \mathbf{e}_{x'_i} \in \mathbb{F}^{\sqrt{N}}$ be the x'_i-th standard unit vector. We reduce f to Aux_N^k:

$$f(x_1, \ldots, x_{2k}) = \langle \mathbf{F}, \mathbf{x}_1 \otimes \ldots \otimes \mathbf{x}_{2k} \rangle$$

where $\mathbf{F}$ is the truth-table of f. For the remainder of the section, it is thus sufficient to construct a PSM protocol for Aux_N^k.

For each non-empty $\Omega \subseteq [2k]$, our protocol will sample a random dimension-$|\Omega|$ tensor $\mathbf{R}_\Omega \in \mathcal{R}^{(\sqrt{N})^{|\Omega|}}$ from the common random string[1]. Define $\bar{\mathbf{X}}_\Omega := \mathbf{R}_\Omega + \bigotimes_{i \in \Omega} \mathbf{x}_i$. E.g., $\bar{\mathbf{X}}_{\{2\}} := \mathbf{R}_{\{2\}} + \mathbf{x}_2$, $\bar{\mathbf{X}}_{\{3,4\}} := \mathbf{R}_{\{3,4\}} + \mathbf{x}_3 \otimes \mathbf{x}_4$.

Within the communication complexity budget $O(N^{\frac{k-1}{2}})$, we can let the referee learn $\bar{\mathbf{X}}_\Omega$ for all Ω such that $|\Omega| \leq k-1$ (more details in Section 3.2). The referee does not learn extra information as $\bar{\mathbf{X}}_\Omega$ is one-time padded by $\mathbf{R}_\Omega$. For example when $k = 4$, we can let the referee learn tensors $\bar{\mathbf{X}}_{\{1\}}, \bar{\mathbf{X}}_{\{2\}}, \ldots, \bar{\mathbf{X}}_{\{8\}}, \bar{\mathbf{X}}_{\{1,2\}}, \bar{\mathbf{X}}_{\{1,3\}}, \ldots, \bar{\mathbf{X}}_{\{7,8\}}, \bar{\mathbf{X}}_{\{1,2,3\}}, \bar{\mathbf{X}}_{\{1,2,4\}}, \ldots, \bar{\mathbf{X}}_{\{6,7,8\}}$. The referee learns those tensor by having *subsets of the parties* recursively perform PSM ptocols with a smaller number of parties, so that the referee learns the required information. Learning those tensors allows the referee to compute many terms including $\langle \mathbf{F}, \bar{\mathbf{X}}_{\{1,2,3\}} \otimes \bar{\mathbf{X}}_{\{4,5,6\}} \otimes \bar{\mathbf{X}}_{\{7,8\}} \rangle$, which equals to the

[1] A note on the randomness complexity: The final protocol uses $\mathbf{R}_\Omega$ only if $|\Omega| \leq k-1$.

sum of the following 8 terms,

$$
\begin{aligned}
&\langle \mathbf{F}, \bar{\mathbf{X}}_{\{1,2,3\}} \otimes \bar{\mathbf{X}}_{\{4,5,6\}} \otimes \bar{\mathbf{X}}_{\{7,8\}} \rangle \\
&= \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{x}_4 \otimes \mathbf{x}_5 \otimes \mathbf{x}_6 \otimes \mathbf{x}_7 \otimes \mathbf{x}_8 \rangle \\
&\quad + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{x}_4 \otimes \mathbf{x}_5 \otimes \mathbf{x}_6 \otimes \mathbf{R}_{\{7,8\}} \rangle \\
&\quad + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{R}_{\{4,5,6\}} \otimes \mathbf{x}_7 \otimes \mathbf{x}_8 \rangle \\
&\quad + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{R}_{\{4,5,6\}} \otimes \mathbf{R}_{\{7,8\}} \rangle \\
&\quad + \langle \mathbf{F}, \mathbf{R}_{\{1,2,3\}} \otimes \mathbf{x}_4 \otimes \mathbf{x}_5 \otimes \mathbf{x}_6 \otimes \mathbf{x}_7 \otimes \mathbf{x}_8 \rangle \\
&\quad + \langle \mathbf{F}, \mathbf{R}_{\{1,2,3\}} \otimes \mathbf{x}_4 \otimes \mathbf{x}_5 \otimes \mathbf{x}_6 \otimes \mathbf{R}_{\{7,8\}} \rangle \\
&\quad + \langle \mathbf{F}, \mathbf{R}_{\{1,2,3\}} \otimes \mathbf{R}_{\{4,5,6\}} \otimes \mathbf{x}_7 \otimes \mathbf{x}_8 \rangle \\
&\quad + \langle \mathbf{F}, \mathbf{R}_{\{1,2,3\}} \otimes \mathbf{R}_{\{4,5,6\}} \otimes \mathbf{R}_{\{7,8\}} \rangle.
\end{aligned} \tag{6}
$$

Before we continue, let us introduce a few notations to describe the terms appearing in (6). The term (tensor) on the left hand side of the equation will be called a *masked term* (masked tensor). The terms (tensors) on the right hand side of the equation will be called *pure terms* (pure tensors).

Definition (masked tensor & masked term)**.** A masked tensor is a tensor product $\bar{\mathbf{X}}_{\Omega_1} \otimes \ldots \otimes \bar{\mathbf{X}}_{\Omega_t}$[2] such that $\Omega_1, \ldots, \Omega_t$ are disjoint and their union equals $[2k]$. The *shape* of a masked tensor $\bar{\mathbf{X}}_{\Omega_1} \otimes \ldots \otimes \bar{\mathbf{X}}_{\Omega_t}$ is the multiset $\{|\Omega_1|, \ldots, |\Omega_t|\}$. The inner product of a masked tensor and $\mathbf{F}$ is called a masked term.

For any multiset P such that $\mathrm{sum}(P) = 2k$, let $\sum \bar{\mathbf{X}}(P)$ denote the sum of all masked tensors of shape P, let $\sum \langle \mathbf{F}, \bar{\mathbf{X}}(P) \rangle$ denote the sum of all masked terms of shape P. We thus have $\sum \langle \mathbf{F}, \bar{\mathbf{X}}(P) \rangle = \langle \mathbf{F}, \sum \bar{\mathbf{X}}(P) \rangle$.

Definition (pure tensor & pure term)**.** A pure tensor is a tensor product $\mathbf{R}_{\Omega_1} \otimes \ldots \otimes \mathbf{R}_{\Omega_t} \otimes \mathbf{x}_{i_1} \otimes \ldots \otimes \mathbf{x}_{i_w}$ such that $\{i_1, \ldots, i_w\}, \Omega_1, \ldots, \Omega_t$ are disjoint and their union equals $[2k]$. The *shape* of a pure tensor $\mathbf{R}_{\Omega_1} \otimes \ldots \otimes \mathbf{R}_{\Omega_t} \otimes \mathbf{x}_{i_1} \otimes \ldots \otimes \mathbf{x}_{i_w}$ is the multiset $\{|\Omega_1|, \ldots, |\Omega_t|\}$. The inner product of a pure tensor and $\mathbf{F}$ is called a pure term.

For any multiset P such that $\mathrm{sum}(P) \leq 2k$, let $\sum \mathbf{R}(P)$ denote the sum of all pure tensors of shape P, let $\sum \langle \mathbf{F}, \mathbf{R}(P) \rangle$ denote the sum of all pure terms of shape P. We thus have $\sum \langle \mathbf{F}, \mathbf{R}(P) \rangle = \langle \mathbf{F}, \sum \mathbf{R}(P) \rangle$.

The pure terms (pure tensors) can be grouped into 3 natural categories:

target term (target tensor) $\langle \mathbf{F}, \mathbf{x}_1 \otimes \ldots \otimes \mathbf{x}_{2k} \rangle$ is called the target term as it is desired functionality output. The corresponding tensor $\mathbf{x}_1 \otimes \ldots \otimes \mathbf{x}_{2k}$ is called the target tensor.

easy terms (easy tensors) A pure tensor $\mathbf{R}_{\Omega_1} \otimes \ldots \otimes \mathbf{R}_{\Omega_t} \otimes \mathbf{x}_{i_1} \otimes \ldots \otimes \mathbf{x}_{i_w}$ is called an easy tensor if at most $k+1$ out of the $2k$ dimensions are contributed by vector $\mathbf{x}_i$'s (i.e., $w \leq k+1$). The corresponding term is called an easy term. Every easy term admits a PSM protocol with communication complexity no more than $O(\mathrm{poly}(k) \cdot N^{\frac{k-1}{2}})$ field elements (more details in Section 3.2).

hard terms (hard tensors) The rest.

[2]We implicitly exchange the order of indices in tensor product. E.g. when $k = 2$, the masked tensor $\bar{\mathbf{X}}_{\{1,4\}} \otimes \bar{\mathbf{X}}_{\{2,3\}}$ is defined by $(\bar{\mathbf{X}}_{\{1,4\}} \otimes \bar{\mathbf{X}}_{\{2,3\}})[j_1, j_2, j_3, j_4] = \bar{\mathbf{X}}_{\{1,4\}}[j_1, j_4] \cdot \bar{\mathbf{X}}_{\{2,3\}}[j_2, j_3]$.

With this terminology, we can give *an overview of our PSM protocol.* As the referee can learn $\bar{\mathbf{X}}_\Omega$ for all Ω such that $|\Omega| \le k-1$, the referee can compute any masked term of shape P if $\max(P) \le k-1$. As suggested by Equation (6), every masked term is the linear combination of a few pure terms. Ideally, the referee only has to combine some computable masked terms, so that all the hard terms cancel out, resulting a linear combination of the target term and easy terms:

$$\textit{a linear combination of masked terms} = \textit{target term} + \textit{some easy terms}. \tag{7}$$

Once we are in this ideal case, the easy terms can be easily removed by standard techniques, resulting the desired k-party PSM protocol for Aux_N^k. (More details are presented in Section 3.2.) Therefore, the task is reduced to a linear algebra problem: *is the target term (resp. tensor) spanned by the referee-computable masked terms (resp. tensors) and easy terms (resp. tensors)?*

When solving such linear algebra problem, it is fair to assume that the solution is symmetric. (Otherwise, assume that a solution that looks like (7) is asymmetric, it can be symmetrized by applying the symmetric sum on both sides.)

We have defined the (symmetric) sum of terms or tensors of the same shape. For example when $k=4$, $\sum\bar{\mathbf{X}}(3,3,2)$ is defined as the sum of all masked tensors $\bar{\mathbf{X}}_{\Omega_1} \otimes \bar{\mathbf{X}}_{\Omega_2} \otimes \bar{\mathbf{X}}_{\Omega_3}$ such that the multiset $\{|\Omega_1|, |\Omega_2|, |\Omega_3|\}$ equals $\{3,3,2\}$, i.e.

$$\begin{aligned}\sum\bar{\mathbf{X}}(3,3,2) := {} & \bar{\mathbf{X}}_{\{1,2,3\}} \otimes \bar{\mathbf{X}}_{\{4,5,6\}} \otimes \bar{\mathbf{X}}_{\{7,8\}} + \bar{\mathbf{X}}_{\{1,2,3\}} \otimes \bar{\mathbf{X}}_{\{4,5,7\}} \otimes \bar{\mathbf{X}}_{\{6,8\}} \\ & + \bar{\mathbf{X}}_{\{1,2,3\}} \otimes \bar{\mathbf{X}}_{\{4,5,8\}} \otimes \bar{\mathbf{X}}_{\{5,6\}} + \bar{\mathbf{X}}_{\{1,2,3\}} \otimes \bar{\mathbf{X}}_{\{4,6,7\}} \otimes \bar{\mathbf{X}}_{\{5,8\}} \\ & + \ldots + \bar{\mathbf{X}}_{\{3,4,5\}} \otimes \bar{\mathbf{X}}_{\{6,7,8\}} \otimes \bar{\mathbf{X}}_{\{1,2\}}.\end{aligned}$$

Let's revisit Equation (6),

$$\begin{aligned}&\underbrace{\langle \mathbf{F}, \bar{\mathbf{X}}_{\{1,2,3\}} \otimes \bar{\mathbf{X}}_{\{4,5,6\}} \otimes \bar{\mathbf{X}}_{\{7,8\}} \rangle}_{\text{a masked term of shape } \{3,3,2\}} \\ &= \underbrace{\langle \mathbf{F}, \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_8 \rangle}_{\text{a pure term of shape } \{\}} + \underbrace{\langle \mathbf{F}, \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_6 \otimes \mathbf{R}_{\{7,8\}} \rangle}_{\text{a pure term of shape } \{2\}} \\ &+ \underbrace{\langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{R}_{\{4,5,6\}} \otimes \mathbf{x}_7 \otimes \mathbf{x}_8 \rangle + \langle \mathbf{F}, \mathbf{R}_{\{1,2,3\}} \otimes \mathbf{x}_4 \otimes \cdots \otimes \mathbf{x}_8 \rangle}_{\text{pure terms of shape } \{3\}} \\ &+ \underbrace{\langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{R}_{\{4,5,6\}} \otimes \mathbf{R}_{\{7,8\}} \rangle + \langle \mathbf{F}, \mathbf{R}_{\{1,2,3\}} \otimes \mathbf{x}_4 \otimes \mathbf{x}_5 \otimes \mathbf{x}_6 \otimes \mathbf{R}_{\{7,8\}} \rangle}_{\text{pure terms of shape } \{3,2\}} \\ &+ \underbrace{\langle \mathbf{F}, \mathbf{R}_{\{1,2,3\}} \otimes \mathbf{R}_{\{4,5,6\}} \otimes \mathbf{x}_7 \otimes \mathbf{x}_8 \rangle}_{\text{a pure term of shape } \{3,3\}} + \underbrace{\langle \mathbf{F}, \mathbf{R}_{\{1,2,3\}} \otimes \mathbf{R}_{\{4,5,6\}} \otimes \mathbf{R}_{\{7,8\}} \rangle}_{\text{a pure term of shape } \{3,3,2\}}.\end{aligned}$$

By applying a symmetric sum on both sides, we get

$$\begin{aligned}\sum\langle \mathbf{F}, \bar{\mathbf{X}}(3,3,2) \rangle = {} & \underbrace{280 \cdot \sum\langle \mathbf{F}, \mathbf{R}() \rangle}_{\text{the target term}} + \underbrace{10 \cdot \sum\langle \mathbf{F}, \mathbf{R}(2) \rangle}_{\text{hard pure terms}} \\ & + \underbrace{10 \cdot \sum\langle \mathbf{F}, \mathbf{R}(3) \rangle + \sum\langle \mathbf{F}, \mathbf{R}(3,2) \rangle + \sum\langle \mathbf{F}, \mathbf{R}(3,3) \rangle + \sum\langle \mathbf{F}, \mathbf{R}(3,3,2) \rangle}_{\text{easy pure terms}}.\end{aligned}$$

As another example of the symmetric sum of masked term that the referee can compute,

$$\sum\langle \mathbf{F}, \bar{\mathbf{X}}(2,2,2,2)\rangle = \underbrace{105 \cdot \sum\langle \mathbf{F}, \mathbf{R}()\rangle}_{\text{target term}} + \underbrace{15 \cdot \sum\langle \mathbf{F}, \mathbf{R}(2)\rangle}_{\text{hard pure terms}} + \underbrace{3 \cdot \sum\langle \mathbf{F}, \mathbf{R}(2,2)\rangle + \sum\langle \mathbf{F}, \mathbf{R}(2,2,2)\rangle + \sum\langle \mathbf{F}, \mathbf{R}(2,2,2,2)\rangle}_{\text{easy pure terms}}.$$

By carefully combining the above two equations, we get

$$3 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(3,3,2)\rangle - 2 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(2,2,2,2)\rangle = 630 \cdot \sum\langle \mathbf{F}, \mathbf{R}()\rangle + \text{easy terms}, \tag{8}$$

which induces a 4-party PSM whose communication complexity is $O(N^{3/2})$, if we let $\mathbb{F}$ to be any field in which $630 \neq 0$. (Section 3.2 explains how Equation (8) implies a 4-party PSM with desired communication complexity.)

In the general k-party case, for each legit shape P of masked term (i.e., P is a multiset consisting of positive integers and $\text{sum}(P) = 2k$),

$$\sum\langle \mathbf{F}, \bar{\mathbf{X}}(P)\rangle = \sum_{Q \subseteq P} \alpha(Q) \cdot \sum\langle \mathbf{F}, \mathbf{R}(P \setminus Q)\rangle, \tag{9}$$

where $P \setminus Q$ is the multiset subtraction and

$$\alpha(Q) := \frac{(\text{sum}(Q))!}{\prod_{i \in Q} i! \cdot \prod_{m \in \mathbb{Z}^+} (\text{number of } m\text{'s in } Q)!} \tag{10}$$

is the following combinatoric number: $\alpha(Q)$ is the number of ways to partition $\text{sum}(Q)$ distinct elements into some unordered subsets $S_1, \ldots, S_t$ such that $Q = \{|S_1|, \ldots, |S_t|\}$. Equations (9), (10) are proved in Appendix A.

3.2 The Induced PSM Protocol

In order to develop the previous section smoothly, we skipped some technique details in Section 3.1. In this section, we will show how to construct a k-party PSM protocol assuming that the target term is spanned by referee-computable masked terms and easy pure terms.

By our assumption, there are referee-computable masked terms $\bar{\mathbf{X}}^{(1)}, \ldots, \bar{\mathbf{X}}^{(t)}$, easy pure terms $\mathbf{R}^{(1)}, \ldots, \mathbf{R}^{(s)}$, and coefficients $a_1, \ldots a_t, b_1, \ldots, b_s \in \mathbb{F}$ such that

$$\langle \mathbf{F}, \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_{2k}\rangle = \sum_{j=1}^{t} a_j \bar{\mathbf{X}}^{(j)} + \sum_{j=1}^{s} b_j \mathbf{R}^{(j)}. \tag{11}$$

A k-party PSM for f, together with its correctness and security, is yielded by the following facts:

- Fact I: $\sum_{j=1}^{s} b_j \mathbf{R}^{(j)}$ and $\bar{\mathbf{X}}_\Omega$ for all $0 < |\Omega| \leq k-1$ form a randomized encoding of $\langle \mathbf{F}, \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_{2k}\rangle$. That is, they contain the sufficient information to recover $\langle \mathbf{F}, \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_{2k}\rangle$, and they are garbled with additional randomness so that no other information can be recovered.
- Fact II: For every $\Omega \subseteq [2k]$ such that $0 < |\Omega| \leq k-1$, there is a PSM protocol for $\bar{\mathbf{X}}_\Omega$ with communication complexity $\text{poly}(k) \cdot N^{\frac{k-1}{2}}$ field elements.

- Fact III: There is a PSM protocol for $\sum_{j=1}^{s} b_j \mathbf{R}^{(j)}$ with communication complexity $\text{poly}(k) \cdot s \cdot N^{\frac{k-1}{2}}$ field elements.

The k-party PSM for f works as the follows: For each $\Omega \subseteq [2k]$ such that $0 < |\Omega| \leq k-1$, use the PSM guaranteed by Fact II to reveal $\bar{\mathbf{X}}_\Omega$ to the referee. Use the PSM guaranteed by Fact III to reveal $\sum_{j=1}^{s} b_j \mathbf{R}^{(j)}$ to the referee. Then Fact I allows the referee to compute the output from Equation (11).

Proof of Fact I. Equation (11) shows that $\langle \mathbf{F}, \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_{2k} \rangle$ can be computed from the encoding. Moreover, the distribution of the encoding is perfectly simulatable: The joint distribution of tensors $\bar{\mathbf{X}}_\Omega$ for $0 < |\Omega| \leq k-1$ is uniform distribution, as they are independently one-time padded. Then the value of $\sum_{j=1}^{s} b_j \mathbf{R}^{(j)}$ is uniquely determined by equation (11).

Proof of Fact II. Each coordinate of $\mathbf{X}_\Omega$ is defined as

$$\bar{\mathbf{X}}_{\{j_1,\ldots,j_t\}}[i_1,\ldots,i_t] = \mathbf{R}_{\{j_1,\ldots,j_t\}}[i_1,\ldots,i_t] + \mathbf{x}_{j_1}[i_1] \cdot \ldots \cdot \mathbf{x}_{j_t}[i_t],$$

which is an arithmetic formula of size $O(k)$. Thus each coordinate has a PSM protocol with communication complexity $\text{poly}(k)$ field elements [IK00].

Proof of Fact III. Sample random $c_1, \ldots, c_s \in \mathbb{F}$ from the common random string such that $c_1 + \ldots + c_s = 0$. Then it's sufficient to construct a PSM protocol for computing $b_j \mathbf{R}^{(j)} + c_j$ for each j. Say this easy pure term $\mathbf{R}^{(j)}$ is $\langle \mathbf{F}, \mathbf{R}_{\Omega_1} \otimes \ldots \otimes \mathbf{x}_{i_1} \otimes \ldots \otimes \mathbf{x}_{i_w} \rangle$. By our definition of an easy term, $w \leq k+1$. There exists a special party, such that the other parties hold at most $k-1$ of $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_w}$. When $w = k+1$, the special party is the one who holds two of $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_w}$ (the existence is guaranteed by the pigeonhole principle). W.l.o.g. assume that the other parties hold $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{w'}}$ such that $w' \leq k-1$. Then the special party knows a dimension-w' tensor $\mathbf{G}$ (which is determined by its input and $b_j, \mathbf{R}_{\Omega_1}, \mathbf{R}_{\Omega_2}, \ldots$) such that

$$b_j \mathbf{R}^{(j)} + c_j = \langle \mathbf{G}, \mathbf{x}_{i_1} \otimes \ldots \otimes \mathbf{x}_{i_{w'}} \rangle + c_j,$$

which admits a PSM protocol (presented in Section B.1) with communication complexity $O(\text{poly}(k) \cdot N^{w'/2})$ field elements.

3.3 When k is Small

As shown in Section 3.1, to construct PSM protocol for Aux_N^k with communication complexity $O_k(N^{\frac{k-1}{2}})$, it is sufficient to prove the target term is spanned by the referee-computable masked terms and easy pure terms. In this section, we verify the condition holds for all $k \leq 20$, which proves the first bullet of Theorem 3.1. However, we do not have a general construction of such linenar systems of equations for an arbitrary k.

The case when $k = 2$ was solved by [BIKK14]. Our framework yields the same protocol from

$$\sum \langle \mathbf{F}, \bar{\mathbf{X}}(1,1,1,1) \rangle = \sum \langle \mathbf{F}, \mathbf{R}() \rangle + \text{easy terms}.$$

The case when $k = 3$ was solved by [BKN18]. Our framework yields a similar protocol from

$$\sum \langle \mathbf{F}, \bar{\mathbf{X}}(2,2,2) \rangle = \sum \langle \mathbf{F}, \mathbf{R}() \rangle + \text{easy terms}.$$

The case when $k = 4$ is solved in section 3.1.

For $k = 5$, consider the following two masked terms,

$$\sum\langle \mathbf{F}, \bar{\mathbf{X}}(4,4,2)\rangle = 1575 \cdot \sum\langle \mathbf{F}, \mathbf{R}()\rangle + 35 \cdot \sum\langle \mathbf{F}, \mathbf{R}(2)\rangle + \text{easy terms},$$
$$\sum\langle \mathbf{F}, \bar{\mathbf{X}}(4,2,2,2)\rangle = 3150 \cdot \sum\langle \mathbf{F}, \mathbf{R}()\rangle + 210 \cdot \sum\langle \mathbf{F}, \mathbf{R}(2)\rangle + \text{easy terms}.$$

We have $6 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(4,4,2)\rangle - \sum\langle \mathbf{F}, \bar{\mathbf{X}}(4,2,2,2)\rangle = 6300 \cdot \sum\langle \mathbf{F}, \mathbf{R}()\rangle + \text{easy terms}$, which induces a 5-party PSM with communication complexity $O(N^2)$.

For $k = 6$, consider the following masked terms

$$\begin{bmatrix} \sum\langle \mathbf{F}, \bar{\mathbf{X}}(5,4,3)\rangle \\ \sum\langle \mathbf{F}, \bar{\mathbf{X}}(4,4,4)\rangle \\ \sum\langle \mathbf{F}, \bar{\mathbf{X}}(3,3,3,3)\rangle \end{bmatrix} = \begin{bmatrix} 27720 & 126 & 56 \\ 5775 & & 35 \\ 15400 & 280 & \end{bmatrix} \begin{bmatrix} \sum\langle \mathbf{F}, \mathbf{R}()\rangle \\ \sum\langle \mathbf{F}, \mathbf{R}(3)\rangle \\ \sum\langle \mathbf{F}, \mathbf{R}(4)\rangle \end{bmatrix} + \text{easy terms}$$

Therefore, $100 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(5,4,3)\rangle - 160 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(4,4,4)\rangle - 45 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(3,3,3,3)\rangle = 1155000 \cdot \sum\langle \mathbf{F}, \mathbf{R}()\rangle + \text{easy terms}$, which induces a 6-party PSM with communication complexity $O(N^{2.5})$.

For $k = 7$, consider the following masked terms

$$\begin{bmatrix} \sum\langle \mathbf{F}, \bar{\mathbf{X}}(4,4,4,2)\rangle \\ \sum\langle \mathbf{F}, \bar{\mathbf{X}}(6,6,2)\rangle \\ \sum\langle \mathbf{F}, \bar{\mathbf{X}}(6,4,4)\rangle \end{bmatrix} = \begin{bmatrix} 525525 & 5775 & 1575 \\ 42042 & 462 & \\ 105105 & & 210 \end{bmatrix} \begin{bmatrix} \sum\langle \mathbf{F}, \mathbf{R}()\rangle \\ \sum\langle \mathbf{F}, \mathbf{R}(2)\rangle \\ \sum\langle \mathbf{F}, \mathbf{R}(4)\rangle \end{bmatrix} + \text{easy terms}$$

Therefore, $14 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(4,4,4,2)\rangle - 175 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(6,6,2)\rangle - 105 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(6,4,4)\rangle = -11036025 \cdot \sum\langle \mathbf{F}, \mathbf{R}()\rangle + \text{easy terms}$, which induces a 7-party PSM with communication complexity $O(N^3)$.

For larger k, we wrote a simple program[3] to check if the target term can be spanned by referee-computable masked terms and easy terms. For simplicity, our program requires specifying the finite field in advance. Our program verifies that the framework yields a PSM protocol with c.c. $O(N^{\frac{k-1}{2}})$ for every $k \leq 20$. For example when $k = 20$, our program found:

$$\sum\langle \mathbf{F}, \mathbf{R}()\rangle = 2895 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,19,2)\rangle + 1902 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,17,4)\rangle + 2843 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,16,5)\rangle + 1025 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,16,3,2)\rangle + 691 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,15,6)\rangle + 2507 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,15,4,2)\rangle + 1923 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,14,7)\rangle + 1836 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,14,5,2)\rangle + 2385 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,13,8)\rangle + 2073 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,13,6,2)\rangle + 1312 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,12,9)\rangle + 2963 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,12,7,2)\rangle + 568 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,11,10)\rangle + 975 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,11,8,2)\rangle + 2445 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,10,9,2)\rangle + 2047 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,9,8,4)\rangle + 318 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,9,8,2,2)\rangle + 2118 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,9,6,6)\rangle + 2189 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,9,6,4,2)\rangle + 1271 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,9,6,2,2,2)\rangle + 1557 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,9,4,4,4)\rangle + 2482 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,9,4,4,2,2)\rangle + 173 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,9,4,2,2,2,2)\rangle + 1943 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(19,9,2,2,2,2,2,2)\rangle + 29 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,18,4)\rangle + 1247 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,17,5)\rangle + 1768 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,17,3,2)\rangle + 2735 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,16,6)\rangle + 416 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,16,4,2)\rangle + 1009 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,15,7)\rangle + 130 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,15,5,2)\rangle + 138 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,14,8)\rangle + 52 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,14,6,2)\rangle + 2661 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,13,9)\rangle + 26 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,13,7,2)\rangle + 731 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,12,10)\rangle + 16 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,12,8,2)\rangle + 145 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,11,11)\rangle + 12 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,11,9,2)\rangle + 818 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,10,8,4)\rangle + 1728 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,10,8,2,2)\rangle + 2676 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,10,6,6)\rangle + 1533 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,10,6,4,2)\rangle + 2490 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,10,6,2,2,2)\rangle + 760 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,10,4,4,4)\rangle + 747 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,10,4,4,2,2)\rangle + 2752 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,10,4,2,2,2,2)\rangle + 83 \cdot \sum\langle \mathbf{F}, \bar{\mathbf{X}}(18,10,2,2,2,2,2,2)\rangle + \text{easy terms} \mod 3001$$

which induces a PSM protocol with c.c. $O(N^{9.5})$.

3.4 When $k+1$ is a Prime Power

As shown in Section 3.1, to construct PSM protocol for Aux_N^k with communication complexity $O_k(N^{\frac{k-1}{2}})$, it is sufficient to prove the target term is spanned by the referee-computable masked terms and easy pure terms. In this section, we prove that the condition holds for all k such that $k+1$ is a prime power, which proves the second bullet of Theorem 3.1.

When $k+1$ is a prime p or a prime power p^e, we obtain a simple k-party PSM, by doing computations in the finite field $\mathbb{F}_p$.

[3] The source code can be downloaded from https://github.com/tianren/psm.

Proof. Consider the symmetric sum of all masked terms of shape $\{k-1, 1, \ldots, 1\}$

$$\begin{aligned}
&\sum\langle \mathbf{F}, \bar{\mathbf{X}}(k-1, \underbrace{1, \ldots, 1}_{k+1 \text{ 1's}})\rangle \\
&= \sum_{i=0}^{k+1} \alpha(k-1, \underbrace{1, \ldots, 1}_{k+1-i \text{ 1's}}) \cdot \sum\langle \mathbf{F}, \mathbf{R}(\underbrace{1, \ldots, 1}_{i \text{ 1's}})\rangle \\
&\quad + \sum_{i=0}^{k+1} \alpha(\underbrace{1, \ldots, 1}_{k+1-i \text{ 1's}}) \cdot \sum\langle \mathbf{F}, \mathbf{R}(k-1, \underbrace{1, \ldots, 1}_{i \text{ 1's}})\rangle \qquad (12) \\
&= \alpha(k-1, \underbrace{1, \ldots, 1}_{k+1 \text{ 1's}}) \cdot \sum\langle \mathbf{F}, \mathbf{R}()\rangle \\
&\quad + \sum_{i=1}^{k-2} \alpha(k-1, \underbrace{1, \ldots, 1}_{k+1-i \text{ 1's}}) \cdot \sum\langle \mathbf{F}, \mathbf{R}(\underbrace{1, \ldots, 1}_{i \text{ 1's}})\rangle + \text{easy terms.}
\end{aligned}$$

(Recall that a pure term of shape P is easy iff $\text{sum}(P) \geq k-1$.)

W.l.o.g. assume $k > 2$. By definition, $\alpha(k-1, \underbrace{1, \ldots, 1}_{t \text{ 1's}}) = \binom{k-1+t}{k-1}$. Lemma 3.3 shows that $\alpha(k-1, \underbrace{1, \ldots, 1}_{k+1 \text{ 1's}}) = \binom{2k}{k-1} \equiv 1 \mod p$, while $\alpha(k-1, \underbrace{1, \ldots, 1}_{k+1-i \text{ 1's}}) = \binom{2k-i}{k-1}$ is a multiple of p for all $1 \leq i \leq k-2$. Therefore,

$$\sum\langle \mathbf{F}, \bar{\mathbf{X}}(k-1, \underbrace{1, \ldots, 1}_{k+1 \text{ 1's}})\rangle = \sum\langle \mathbf{F}, \mathbf{R}()\rangle + \text{easy terms} \mod p,$$

which induces a k-party PSM protocol with c.c. $O_k(N^{\frac{k-1}{2}})$. □ □

Lemma 3.2. *For any prime p and positive integer e, $\binom{p^e}{t}$ is a multiple of p for all $0 < t < p^e$.*

Proof.

$$\binom{p^e}{t} = \frac{p^e}{t} \cdot \binom{p^e - 1}{t-1}.$$

□

Lemma 3.3. *For any prime p and positive integer e, binomial coefficient $\binom{p^e+t}{p^e-2}$ is a multiple of p for all $0 \leq t \leq p^e - 3$, while binomial coefficient $\binom{2p^e-2}{p^e-2} \equiv 1 \mod p$.*

Proof. For every $0 \leq t \leq p^e - 3$,

$$\binom{p^e + t}{p^e - 2} = \sum_{j=0}^{t} \binom{t}{j} \underbrace{\binom{p^e}{p^e - 2 - j}}_{\text{multiple of } p}$$

is a multiple of p. While

$$\binom{2p^e - 2}{p^e - 2} = \sum_{j=0}^{p^e-3} \binom{p^e - 2}{j} \underbrace{\binom{p^e}{p^e - 2 - j}}_{\text{multiple of } p} + \binom{p^e - 2}{p^e - 2}\binom{p^e}{0} \equiv 1 \mod p.$$

□

4 Unbalanced 2-party PSM Protocols

The two parties in 2-party PSM are conventionally called Alice and Bob. Let $x \in [N]$ denote Alice's and $y \in [N]$ denote Bob's input. In this section, we show that every functionality $f : [N] \times [N] \to \{0,1\}$ admits a 2-party PSM protocol, where Alice sends $O(N^\eta)$ bits and Bob sends $O(N^{1-\eta})$ bits.

Theorem 4.1. *For any functionality $f : [N] \times [N] \to \{0,1\}$, and any $\eta = d/k$ such that d, k are integers and $0 < d < k \leq 20$, there is a 2-party PSM protocol for f with unbalanced communication complexity $O(N^\eta), O(N^{1-\eta})$.*

In this section, we prove a stronger statement. Let $\mathbb{F}$ be a finite field, consider the following auxiliary 2-party functionality $\mathsf{Aux}^2_{k,N}$:

2-party functionality $\mathsf{Aux}^2_{k,N}$

- Alice has input $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbb{F}^{\sqrt[k]{N}}$
- Bob has input $\mathbf{y}_1, \ldots, \mathbf{y}_k \in \mathbb{F}^{\sqrt[k]{N}}$
- The output is $\langle \mathbf{F}, \mathbf{x}_1 \otimes \ldots \otimes \mathbf{x}_k \otimes \mathbf{y}_1 \otimes \ldots \otimes \mathbf{y}_k \rangle$, where $\mathbf{F}$ is public and fixed

A PSM protocol for $\mathsf{Aux}^2_{k,N}$ implies a PSM for $f : [N] \times [N] \to \{0,1\}$ with the same communication complexity of each party. The reduction consists of having $\mathbf{F}$ be the truth table of f.

We present a framework for the construction of 2-party PSM protocols for $\mathsf{Aux}^2_{k,N}$, where Alice sends $O_\eta(N^\eta)$ bits and Bob sends $O_\eta(N^{1-\eta})$ bits, for all $\eta \in \{\frac{1}{k}, \ldots, \frac{k-1}{k}\}$. Similar to the framework in Section 3, the framework in this section also reduces the problem to a system of linear equations. A solution of the system implies a 2-party PSM protocol with the desired communication complexity. By verifying with a computer, we find that our framework works well for all η whose denominator is no larger than 20. Backed by those observations, we believe that our framework allows for a smooth trade-off between the communication complexity of Alice and Bob:

Conjecture 2. For any functionality $f : [N] \times [N] \to \{0,1\}$, and any $0 < \eta < 1$, there is a 2-party PSM protocol for f with unbalanced communication complexity $O_\eta(N^\eta), O_\eta(N^{1-\eta})$.

Organization Section 4.1 presents our framework for constructing multi-party PSM, introduces new notations, and gives as a concrete example a 2-party PSM with communication $O(N^{1/3}), O(N^{2/3})$. The following Sections 4.2, 4.3 are independent. Section 4.2 provides more technical detail of the PSM protocols yielded by our framework. Section 4.3 shows how the framework works for small k.

4.1 A Framework for 2-party PSM

Consider a rational $\eta = \frac{d}{k} \in (0,1)$. Let $\mathbb{F}$ be a finite commutative ring that we will fix later. All the operations are within ring $\mathbb{F}$ unless otherwise specified.

As mentioned in the beginning of Section 4, there is an non-interactive reduction from the functionality $f : [N] \times [N] \to \{0,1\}$ to functionality $\mathsf{Aux}^2_{k,N}$. The reduction works as follows: Let $x, y \in [N]$ be the input of Alice and Bob respectively. Evenly divide x into $x_1, \ldots, x_k \in [\sqrt[k]{N}]$, similarly divide y into $y_1, \ldots, y_k \in [\sqrt[k]{N}]$. For each $j \in [k]$, let $\mathbf{x}_j := \mathbf{e}_{x_j} \in \mathbb{F}^{\sqrt{N}}$ be the x_j-th

standard unit vector. Similarly let $\mathbf{y}_i := \mathbf{e}_{y_i} \in \mathbb{F}^{\sqrt[k]{N}}$ for every $i \in [k]$. The functionality f can be reduced to $\mathsf{Aux}^2_{k,N}$ by doing:

$$f(x_1, \ldots, x_k, y_1, \ldots, y_k) = \langle \mathbf{F}, \mathbf{x}_1 \otimes \ldots \otimes \mathbf{x}_k \otimes \mathbf{y}_1 \otimes \ldots \otimes \mathbf{y}_k \rangle.$$

where $\mathbf{F}$ is the truth-table of f. For the remainder of the section, it is thus sufficient to construct a PSM protocol for $\mathsf{Aux}^2_{k,N}$.

For every $\Omega \subseteq [k]$, our protocol will sample random $\mathbf{R}_\Omega, \mathbf{S}_\Omega \in \mathbb{F}^{(\sqrt[k]{N})^{|\Omega|}}$ from the common random string. Let $\bar{\mathbf{X}}_\Omega := \mathbf{R}_\Omega + \bigotimes_{i\in\Omega} \mathbf{x}_i$ and $\bar{\mathbf{Y}}_\Omega := \mathbf{S}_\Omega + \bigotimes_{i\in\Omega} \mathbf{y}_i$.

As the communication complexity of Alice is $O_\eta(N^{\frac{d}{k}})$, she can send $\bar{\mathbf{X}}_\Omega$ to the referee for every Ω that $|\Omega| \le d$. So far no information is leaked as $\bar{\mathbf{X}}_\Omega$ is one-time padded by $\mathbf{R}_\Omega$. Similarly, Bob can send $\bar{\mathbf{Y}}_\Omega$ for every Ω that $|\Omega| \le k - d$.

There are many meaningful terms that the referee can compute once he receives $(\bar{\mathbf{X}}_\Omega)_{|\Omega|\le d}$ and $(\bar{\mathbf{Y}}_\Omega)_{|\Omega|\le k-d}$. For example, when $\eta = d/k = 1/3$, the referee can compute:

$$\begin{aligned}
&\langle \mathbf{F}, \bar{\mathbf{X}}_{\{1\}} \otimes \bar{\mathbf{X}}_{\{2\}} \otimes \bar{\mathbf{X}}_{\{3\}} \otimes \bar{\mathbf{Y}}_{\{1,2\}} \otimes \bar{\mathbf{Y}}_{\{3\}} \rangle \\
&= \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{y}_3 \rangle \\
&\quad + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{S}_{\{3\}} \rangle \\
&\quad + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{S}_{\{1,2\}} \otimes \mathbf{S}_{\{3\}} \rangle \\
&\quad + \ldots \quad \text{(28 other terms)} \\
&\quad + \langle \mathbf{F}, \mathbf{R}_{\{1\}} \otimes \mathbf{R}_{\{2\}} \otimes \mathbf{R}_{\{3\}} \otimes \mathbf{S}_{\{1,2\}} \otimes \mathbf{S}_{\{3\}} \rangle.
\end{aligned} \tag{13}$$

Before we continue, we have to introduce a few notations. We will define shape, masked tensor, pure tensor, easy & hard tensor, etc., in the same way as in Section 3.1.

Definition (masked tensor & masked term)**.** An Alice-side masked tensor is a tensor product $\bar{\mathbf{X}}_{\Omega_1} \otimes \ldots \otimes \bar{\mathbf{X}}_{\Omega_t}$ such that $\Omega_1, \ldots, \Omega_t$ are disjoint and their union equals $[k]$. The *shape* of an Alice-side masked tensor $\bar{\mathbf{X}}_{\Omega_1} \otimes \ldots \otimes \bar{\mathbf{X}}_{\Omega_t}$ is the multiset $\{|\Omega_1|, \ldots, |\Omega_t|\}$. Bob-side masked tensors are defined symmetrically.

The tensor product of an Alice-side masked tensor and a Bob-side masked tensor is called a *masked tensor*. The inner product of $\mathbf{F}$ and a masked tensor is called a *masked term*.

An Alice-side masked tensor of shape P is referee-computable if $\max(P) \le d$. A Bob-side masked tensor of shape Q is referee-computable if $\max(Q) \le k - d$. An masked tensor (and its corresponding masked term) is called *referee-computable* if it's the tensor product of a referee-computable Alice-side masked tensor and a referee-computable Bob-side masked tensor.

Definition (pure tensor & pure term)**.** An Alice-side pure tensor is a tensor product $\mathbf{R}_{\Omega_1} \otimes \ldots \otimes \mathbf{R}_{\Omega_t} \otimes \mathbf{x}_{i_1} \otimes \ldots \otimes \mathbf{x}_{i_w}$ such that $\{i_1, \ldots, i_w\}, \Omega_1, \ldots, \Omega_t$ are disjoint and their union equals $[k]$. The *shape* of an Alice-side masked tensor $\mathbf{R}_{\Omega_1} \otimes \ldots \otimes \mathbf{R}_{\Omega_t} \otimes \mathbf{x}_{i_1} \otimes \ldots \otimes \mathbf{x}_{i_w}$ is the multiset $\{|\Omega_1|, \ldots, |\Omega_t|\}$. Bob-side pure tensors are defined symmetrically.

The tensor product of an Alice-side pure tensor and a Bob-side pure tensor is called a *pure tensor*. The inner product of a pure tensor and $\mathbf{F}$ is called a *pure term*.

For any legit shape, let $\sum\mathbf{R}(P)$ denote the sum of all Alice-side pure tensor whose shape is P. Similarly, define Bob-side pure tensor sum $\sum\mathbf{S}(P)$.

Let's go back to the example when $\eta = 1/3$: examine the pure terms on the right side of equation (13), and check which of them has a 2-party PSM with communication complexity $O(N^{\frac{1}{3}}), O(N^{\frac{2}{3}})$.

- The term $\langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{y}_3 \rangle$ is the desired functionality.
- The term $\langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{S}_{\{1,2\}} \otimes \mathbf{y}_3 \rangle$ has a PSM protocol with communication complexity $O(N^{\frac{1}{3}})$. Because Alice knows a vector $\mathbf{g}$ (which is determined by $\mathbf{F}$, Alice's input and randomness $(\mathbf{R}_\Omega)_\Omega, (\mathbf{S}_\Omega)_\Omega$) such that $\langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{S}_{\{1,2\}} \otimes \mathbf{y}_3 \rangle = \langle \mathbf{g}, \mathbf{y}_3 \rangle$.
- The term $\langle \mathbf{F}, \mathbf{S}_{\{1\}} \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{y}_3 \rangle$ admits a PSM protocol with unbalanced communication complexity $O(N^{\frac{1}{3}})$, $O(N^{\frac{2}{3}})$. Because Bob knows a dimension-2 tensor $\mathbf{G}$ such that $\langle \mathbf{F}, \mathbf{S}_{\{1\}} \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{y}_3 \rangle = \langle \mathbf{x}_2 \otimes \mathbf{x}_3, \mathbf{G} \rangle$. (This PSM is presented in Section B.2.)

The discussion above hints at the right classification of pure tensors.

target tensor The only Alice-side target tensor is $\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_k$. The only Bob-side target tensor is $\mathbf{y}_1 \otimes \cdots \otimes \mathbf{y}_k$. The only target tensor is $\mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_k \otimes \mathbf{y}_1 \otimes \cdots \otimes \mathbf{y}_k$.

easy tensor An Alice-side pure tensor of shape P is called easy if $\mathrm{sum}(P) \geq d$. A Bob-side pure tensor of shape Q is called easy if $\mathrm{sum}(Q) \geq k - d$. A pure tensor $\mathbf{R} \otimes \mathbf{S}$ is called easy if either $\mathbf{R}$ or $\mathbf{S}$ is easy.

hard tensor The rest.

The inner product of $\mathbf{F}$ and a target/easy/hard tensor is called a target/easy/hard term.

Then, equation (13) can be rewritten by grouping and ignoring the easy terms:

$$\begin{aligned}&\langle \mathbf{F}, \bar{\mathbf{X}}_{\{1\}} \otimes \bar{\mathbf{X}}_{\{2\}} \otimes \bar{\mathbf{X}}_{\{3\}} \otimes \bar{\mathbf{Y}}_{\{1,2\}} \otimes \bar{\mathbf{Y}}_{\{3\}} \rangle \\ &= \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{y}_3 \rangle \\ &\quad + \langle \mathbf{F}, \mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \mathbf{x}_3 \otimes \mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{S}_{\{3\}} \rangle + \text{easy terms}\end{aligned}.$$

By a symmetric sum, we get

$$\begin{aligned}&\langle \mathbf{F}, \textstyle\sum \bar{\mathbf{X}}(1,1,1) \otimes \sum \bar{\mathbf{Y}}(2,1) \rangle \\ &= 3 \cdot \underbrace{\langle \mathbf{F}, \textstyle\sum \mathbf{R}() \otimes \sum \mathbf{S}() \rangle}_{\text{target}} + \langle \mathbf{F}, \textstyle\sum \mathbf{R}() \otimes \sum \mathbf{S}(1) \rangle + \text{easy terms.}\end{aligned}$$

Similarly, we have decomposed another referee-computable term

$$\begin{aligned}&\langle \mathbf{F}, \textstyle\sum \bar{\mathbf{X}}(1,1,1) \otimes \sum \bar{\mathbf{Y}}(1,1,1) \rangle \\ &= \underbrace{\langle \mathbf{F}, \textstyle\sum \mathbf{R}() \otimes \sum \mathbf{S}() \rangle}_{\text{target}} + \langle \mathbf{F}, \textstyle\sum \mathbf{R}() \otimes \sum \mathbf{S}(1) \rangle + \text{easy terms.}\end{aligned}$$

Combine them to cancel out the hard terms:

$$\begin{aligned}&\langle \mathbf{F}, \textstyle\sum \bar{\mathbf{X}}(1,1,1) \otimes \sum \bar{\mathbf{Y}}(2,1) \rangle - \langle \mathbf{F}, \textstyle\sum \bar{\mathbf{X}}(1,1,1) \otimes \sum \bar{\mathbf{Y}}(1,1,1) \rangle \\ &= 2 \cdot \langle \mathbf{F}, \textstyle\sum \mathbf{R}() \otimes \sum \mathbf{S}() \rangle + \text{easy terms.}\end{aligned}$$

Thus, by setting $\mathbb{F}$ to be any finite field where $2 \neq 0$, the above equation induces a 2-party PSM protocol with unbalanced communication complexity $O(N^{\frac{1}{3}}), O(N^{\frac{2}{3}})$.

In general, a masked term $\langle \mathbf{F}, \sum\bar{\mathbf{X}}(P) \otimes \sum\bar{\mathbf{Y}}(Q)\rangle$ can be decomposed into pure terms by

$$\begin{aligned}
\sum\bar{\mathbf{X}}(P) &= \sum_{P'\subseteq P} \alpha(P')\sum\bar{\mathbf{X}}(P\setminus P'),\\
\sum\bar{\mathbf{Y}}(Q) &= \sum_{Q'\subseteq Q} \alpha(Q')\sum\bar{\mathbf{Y}}(Q\setminus Q'),\\
\langle \mathbf{F}, \sum\bar{\mathbf{X}}(P) \otimes \sum\bar{\mathbf{Y}}(Q)\rangle &= \sum_{\substack{P'\subseteq P\\ Q'\subseteq Q}} \alpha(P')\alpha(Q')\langle \mathbf{F}, \sum\bar{\mathbf{X}}(P\setminus P') \otimes \sum\bar{\mathbf{Y}}(Q\setminus Q')\rangle.
\end{aligned}$$

with the combinatoric number α defined as in Section 3.1. The first two equations are essentially the same as equation (9) and they imply the third equation.

To construct a PSM protocol of the desired unbalanced communication complexity, it is sufficient to show the target term is spanned by the referee-computable masked terms and the easy terms. Namely,

$$\begin{aligned}
&\textit{the target term} = \textit{a linear combination of referee-computable masked terms} +\\
&\quad \textit{a linear combination of easy terms.}
\end{aligned} \tag{14}$$

The details of how this sufficient condition implies a PSM with desired communication complexity is presented in Section 4.2.

This sufficient condition of form (14) is unfortunately too combinatorically hard to use in practice, especially since we are going to use a program to search for the proof for different values of η. There are too many distinct masked terms and pure terms – their number is equal to the number of pairs of legit shapes (P, Q).

Fortunately, we come up with a simpler sufficient condition. A PSM protocol of the desired unbalanced communication complexity exists if both of the following hold:

- The Alice-side target tensor is spanned by referee-computable Alice-side masked tensors and Alice-side easy tensors;
- The Bob-side target tensor is spanned by referee-computable Bob-side masked tensors and Bob-side easy tensors.

The proof is quite straight-forward: Assume the new sufficient condition,

$$\begin{aligned}
&\textit{a linear combination of referee-computable Alice-side masked tensors}\\
&\quad = \sum\mathbf{R}() + \textit{Alice-side easy tensors},\\
&\textit{a linear combination of referee-computable Bob-side masked tensors}\\
&\quad = \sum\mathbf{S}() + \textit{Bob-side easy tensors}.
\end{aligned}$$

The tensor product of the above two equations is

$$\begin{aligned}
&\textit{a linear combination of referee-computable masked tensors}\\
&\quad = \sum\mathbf{R}() \otimes \sum\mathbf{S}() + \textit{a linear combination of easy tensors}.
\end{aligned}$$

Multiplying both sides of the above equation with $\mathbf{F}$ yields the desired sufficient condition of form (14). □

4.2 The Induced PSM Protocol

In order to develop the previous section smoothly, we skipped the technique details on how the condition (14) implies a 2-party PSM of the desired communication complexity. In this section, we will show how to construct such a 2-party PSM protocol assuming that the target term is spanned by referee-computable masked terms and easy pure terms.

By the condition (14), there are referee-computable masked terms $\bar{\mathbf{Z}}^{(1)}, \ldots, \bar{\mathbf{Z}}^{(t)}$, easy pure terms $\mathbf{T}^{(1)}, \ldots, \mathbf{T}^{(s)}$, and coefficients $a_1, \ldots a_t, b_1, \ldots, b_s \in \mathbb{F}$ such that

$$\langle \mathbf{F}, \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_k \otimes \mathbf{y}_1 \otimes \cdots \otimes \mathbf{y}_k \rangle = \sum_{j=1}^{t} a_j \bar{\mathbf{Z}}^{(j)} + \sum_{j=1}^{s} b_j \mathbf{T}^{(j)}. \tag{15}$$

A 2-party PSM for f, together with its correctness and security, is yielded by the following facts:

- Fact I: $\sum_{j=1}^{s} b_j \mathbf{T}^{(j)}$ together with $\bar{\mathbf{X}}_\Omega$ for all $0 < |\Omega| \le d$ and $\bar{\mathbf{Y}}_\Omega$ for all $0 < |\Omega| \le k - d$ form a randomized encoding of the functionality output.
- Fact II: There is a PSM protocol for $\sum_{j=1}^{s} b_j \mathbf{T}^{(j)}$, in which Alice sends $k \cdot s \cdot N^{\frac{d}{k}}$ field elements, Bob sends $k \cdot s \cdot N^{1-\frac{d}{k}}$ field elements.

The 2-party PSM for f works as the follows: For each $\Omega \subseteq [k]$ such that $0 < |\Omega| \le d$, Alice sends $\bar{\mathbf{X}}_\Omega$ to the referee. Symmetrically, for $\Omega \subseteq [k]$ such that $0 < |\Omega| \le k - d$, Bob sends $\bar{\mathbf{Y}}_\Omega$ to the referee. Use the PSM guaranteed by Fact II to reveal $\sum_{j=1}^{s} b_j \mathbf{T}^{(j)}$ to the referee. Then Fact I allows the referee to compute the output from Equation (15).

Proof of Fact I. (Similar to the proof of Fact I in Section 3.2.) Equation (15) shows that $\langle \mathbf{F}, \mathbf{x}_1 \otimes \cdots \otimes \mathbf{x}_k \otimes \mathbf{y}_1 \otimes \cdots \otimes \mathbf{y}_k \rangle$ can be computed from the encoding. Moreover, the distribution of the encoding is perfectly simulatable: The joint distribution of tensors $\bar{\mathbf{X}}_\Omega$ for $0 < |\Omega| \le d$ and $\bar{\mathbf{Y}}_\Omega$ for $0 < |\Omega| \le k - d$ is uniform, as they are independently one-time padded. Then the value of $\sum_{j=1}^{s} b_j \mathbf{T}^{(j)}$ is uniquely determined by Equation (15).

Proof of Fact II. Sample random $c_1, \ldots, c_s \in \mathbb{F}$ from the common random string such that $c_1 + \ldots + c_s = 0$. Then it's sufficient to construct a PSM protocol for computing $b_j \mathbf{T}^{(j)} + c_j$ for each j.

Because $\mathbf{T}^{(j)}$ is an easy term, we have $\mathbf{T}^{(j)} = \langle \mathbf{F}, \mathbf{R}^{(j)} \otimes \mathbf{S}^{(j)} \rangle$, where $\mathbf{R}^{(j)}$ is an Alice-side pure tensor, $\mathbf{S}^{(j)}$ is a Bob-side pure tensor, and either $\mathbf{R}^{(j)}$ is an Alice-side easy tensor, $\mathbf{S}^{(j)}$ is a Bob-side easy tensor. W.l.o.g., assume $\mathbf{R}^{(j)}$ is an Alice-side easy tensor.

Say this Alice-side easy pure term $\mathbf{R}^{(j)}$ is $\mathbf{R}_{\Omega_1} \otimes \ldots \otimes \mathbf{x}_{i_1} \otimes \ldots \otimes \mathbf{x}_{i_w}$. By the definition of an Alice-side easy term, $w \le k - d$. Then Bob knows a dimension-w tensor $\mathbf{G}$ (which is determined by $\mathbf{S}^{(j)}, b_j, \mathbf{R}_{\Omega_1}, \mathbf{R}_{\Omega_2}, \ldots$) such that

$$b_j \mathbf{T}^{(j)} + c_j = \langle \mathbf{G}, \mathbf{x}_{i_1} \otimes \ldots \otimes \mathbf{x}_{i_w} \rangle + c_j,$$

which admits a PSM protocol (presented in Section B.2) in which Alice sends $O(w \cdot N^{1/k})$ field elements, Bob sends $N^{w/k}$ field elements.

4.3 When η has a Small Denominator

Section 4.1 proves a sufficient condition that implies 2-party PSM protocols with the desired unbalanced communication complexity. In this section, we will verify that the sufficient condition holds for all rational $\eta \in (0,1)$ whose denominator is no larger than 20. Theorem 4.1 follows as a consequence.

For $\eta = 1/3$, the 2-party PSM protocol in Section 4.1 is also induced by

$$\begin{aligned} \sum \bar{\mathbf{X}}(1,1,1) &= \sum \mathbf{R}() + \text{Alice-side easy tensors}, \\ \sum \bar{\mathbf{Y}}(2,1) - \sum \bar{\mathbf{Y}}(1,1,1) &= 2 \cdot \sum \mathbf{S}() + \text{Bob-side easy tensors}. \end{aligned}$$

For $\eta = 1/4$, a 2-party PSM protocol with c.c. $O(N^{1/4})$, $O(N^{3/4})$ is induced by

$$\begin{aligned} \sum \bar{\mathbf{X}}(1,1,1,1) &= \sum \mathbf{R}() + \text{Alice-side easy tensors}, \\ \sum \bar{\mathbf{Y}}(1,1,1,1) + 2 \cdot \sum \bar{\mathbf{Y}}(3,1) & \\ + \sum \bar{\mathbf{Y}}(2,2) - \sum \bar{\mathbf{Y}}(2,1,1) &= 6 \cdot \sum \mathbf{S}() + \text{Bob-side easy tensors}. \end{aligned}$$

For $\eta = 1/5$, a 2-party PSM protocol with desired c.c. is induced by

$$\begin{aligned} \sum \bar{\mathbf{X}}(1,1,1,1,1) &= \sum \mathbf{R}() + \text{Alice-side easy tensors}, \\ 6 \cdot \sum \bar{\mathbf{Y}}(4,1) + 2 \cdot \sum \bar{\mathbf{Y}}(3,2) & \\ - 2 \cdot \sum \bar{\mathbf{Y}}(3,1,1) - \sum \bar{\mathbf{Y}}(2,2,1) & \\ + \sum \bar{\mathbf{Y}}(2,1,1,1) - \sum \bar{\mathbf{Y}}(1,1,1,1,1) &= 24 \cdot \sum \mathbf{S}() + \text{Bob-side easy tensors}. \end{aligned}$$

For $\eta = 2/5$, a 2-party PSM protocol with desired c.c. is induced by

$$\begin{aligned} 2 \cdot \sum \bar{\mathbf{X}}(2,2,1) - \sum \bar{\mathbf{X}}(2,1,1,1) &= 20 \cdot \sum \mathbf{R}() + \text{Alice-side easy tensors}, \\ 3 \cdot \sum \bar{\mathbf{Y}}(3,2) + \sum \bar{\mathbf{Y}}(3,1,1) & \\ - \sum \bar{\mathbf{Y}}(2,2,1) - \sum \bar{\mathbf{Y}}(1,1,1,1,1) &= 24 \cdot \sum \mathbf{S}() + \text{Bob-side easy tensors}. \end{aligned}$$

For larger denominators, we wrote a computer program[4] to assist us in the proof. For example, for $\eta = 7/20$, a 2-party PSM with desired c.c. is induced by

$$\begin{aligned} &\Sigma\mathbf{R}() = \text{Alice-side easy tensors} + 18 \cdot \Sigma\bar{\mathbf{X}}(7,7,6) + 10 \cdot \Sigma\bar{\mathbf{X}}(7,7,5,1) + 14 \cdot \Sigma\bar{\mathbf{X}}(7,7,4,2) + 14 \cdot \Sigma\bar{\mathbf{X}}(7,7,4,1,1) + 17 \cdot \Sigma\bar{\mathbf{X}}(7,7,3,3) + 20 \cdot \Sigma\bar{\mathbf{X}}(7,7,3,2,1) + 20 \cdot \\ &\Sigma\bar{\mathbf{X}}(7,7,3,1,1,1) + 10 \cdot \Sigma\bar{\mathbf{X}}(7,7,2,2,2) + 10 \cdot \Sigma\bar{\mathbf{X}}(7,7,2,2,1,1) + 10 \cdot \Sigma\bar{\mathbf{X}}(7,7,2,1,1,1,1) + 10 \cdot \Sigma\bar{\mathbf{X}}(7,7,1,1,1,1,1,1) + 6 \cdot \Sigma\bar{\mathbf{X}}(7,6,6,1) + 19 \cdot \Sigma\bar{\mathbf{X}}(7,6,5,2) + \\ &19 \cdot \Sigma\bar{\mathbf{X}}(7,6,5,1,1) + 21 \cdot \Sigma\bar{\mathbf{X}}(7,6,4,3) + 22 \cdot \Sigma\bar{\mathbf{X}}(7,6,4,2,1) + 22 \cdot \Sigma\bar{\mathbf{X}}(7,6,4,1,1,1) + 7 \cdot \Sigma\bar{\mathbf{X}}(7,6,3,3,1) + 15 \cdot \Sigma\bar{\mathbf{X}}(7,6,3,2,2) + 15 \cdot \Sigma\bar{\mathbf{X}}(7,6,3,2,1,1) + 15 \cdot \\ &\Sigma\bar{\mathbf{X}}(7,6,3,1,1,1,1) + 19 \cdot \Sigma\bar{\mathbf{X}}(7,6,2,2,2,1) + 19 \cdot \Sigma\bar{\mathbf{X}}(7,6,2,2,1,1,1) + 19 \cdot \Sigma\bar{\mathbf{X}}(7,6,2,1,1,1,1,1) + 19 \cdot \Sigma\bar{\mathbf{X}}(7,6,1,1,1,1,1,1,1) \mod 23 \end{aligned}$$

$$\begin{aligned} &\Sigma\mathbf{S}() = \text{Bob-side easy tensors} + 13 \cdot \Sigma\bar{\mathbf{Y}}(13,7) + 20 \cdot \Sigma\bar{\mathbf{Y}}(13,6,1) + 2 \cdot \Sigma\bar{\mathbf{Y}}(13,5,2) + 22 \cdot \Sigma\bar{\mathbf{Y}}(13,5,1,1) + 1 \cdot \Sigma\bar{\mathbf{Y}}(13,4,3) + 17 \cdot \Sigma\bar{\mathbf{Y}}(13,4,2,1) + 3 \cdot \Sigma\bar{\mathbf{Y}}(13,4,1,1,1) + \\ &19 \cdot \Sigma\bar{\mathbf{Y}}(13,3,3,1) + 21 \cdot \Sigma\bar{\mathbf{Y}}(13,3,2,2) + 1 \cdot \Sigma\bar{\mathbf{Y}}(13,3,2,1,1) + 11 \cdot \Sigma\bar{\mathbf{Y}}(13,3,1,1,1,1) + 12 \cdot \Sigma\bar{\mathbf{Y}}(13,2,2,2,1) + 17 \cdot \Sigma\bar{\mathbf{Y}}(13,2,2,1,1,1) + 3 \cdot \Sigma\bar{\mathbf{Y}}(13,2,1,1,1,1,1) + 10 \cdot \\ &\Sigma\bar{\mathbf{Y}}(13,1,1,1,1,1,1,1) + 11 \cdot \Sigma\bar{\mathbf{Y}}(12,8) + 17 \cdot \Sigma\bar{\mathbf{Y}}(12,7,1) + 1 \cdot \Sigma\bar{\mathbf{Y}}(12,6,2) + 11 \cdot \Sigma\bar{\mathbf{Y}}(12,6,1,1) + 17 \cdot \Sigma\bar{\mathbf{Y}}(11,9) + 14 \cdot \Sigma\bar{\mathbf{Y}}(11,8,1) + 6 \cdot \Sigma\bar{\mathbf{Y}}(11,7,2) + 20 \cdot \Sigma\bar{\mathbf{Y}}(11,7,1,1) + \\ &7 \cdot \Sigma\bar{\mathbf{Y}}(11,6,3) + 4 \cdot \Sigma\bar{\mathbf{Y}}(11,6,2,1) + 21 \cdot \Sigma\bar{\mathbf{Y}}(11,6,1,1,1) + 2 \cdot \Sigma\bar{\mathbf{Y}}(10,10) + 4 \cdot \Sigma\bar{\mathbf{Y}}(10,9,1) + 15 \cdot \Sigma\bar{\mathbf{Y}}(10,8,2) + 4 \cdot \Sigma\bar{\mathbf{Y}}(10,8,1,1) + 1 \cdot \Sigma\bar{\mathbf{Y}}(10,7,3) + 17 \cdot \Sigma\bar{\mathbf{Y}}(10,7,2,1) + \\ &3 \cdot \Sigma\bar{\mathbf{Y}}(10,7,1,1,1) + 21 \cdot \Sigma\bar{\mathbf{Y}}(10,6,4) + 8 \cdot \Sigma\bar{\mathbf{Y}}(10,6,3,1) + 4 \cdot \Sigma\bar{\mathbf{Y}}(10,6,2,2) + 21 \cdot \Sigma\bar{\mathbf{Y}}(10,6,2,1,1) + 1 \cdot \Sigma\bar{\mathbf{Y}}(10,6,1,1,1,1) + 20 \cdot \Sigma\bar{\mathbf{Y}}(9,9,2) + 13 \cdot \Sigma\bar{\mathbf{Y}}(9,9,1,1) + 4 \cdot \\ &\Sigma\bar{\mathbf{Y}}(9,8,3) + 22 \cdot \Sigma\bar{\mathbf{Y}}(9,8,2,1) + 12 \cdot \Sigma\bar{\mathbf{Y}}(9,8,1,1,1) + 14 \cdot \Sigma\bar{\mathbf{Y}}(9,7,4) + 13 \cdot \Sigma\bar{\mathbf{Y}}(9,7,3,1) + 18 \cdot \Sigma\bar{\mathbf{Y}}(9,7,2,2) + 14 \cdot \Sigma\bar{\mathbf{Y}}(9,7,2,1,1) + 16 \cdot \Sigma\bar{\mathbf{Y}}(9,7,1,1,1,1) + 11 \cdot \Sigma\bar{\mathbf{Y}}(9,6,5) + \\ &13 \cdot \Sigma\bar{\mathbf{Y}}(9,6,4,1) + 12 \cdot \Sigma\bar{\mathbf{Y}}(9,6,3,2) + 17 \cdot \Sigma\bar{\mathbf{Y}}(9,6,3,1,1) + 20 \cdot \Sigma\bar{\mathbf{Y}}(9,6,2,2,1) + 13 \cdot \Sigma\bar{\mathbf{Y}}(9,6,2,1,1,1) + 5 \cdot \Sigma\bar{\mathbf{Y}}(9,6,1,1,1,1,1) + 19 \cdot \Sigma\bar{\mathbf{Y}}(8,8,4) + 16 \cdot \Sigma\bar{\mathbf{Y}}(8,8,3,1) + \\ &8 \cdot \Sigma\bar{\mathbf{Y}}(8,8,2,2) + 19 \cdot \Sigma\bar{\mathbf{Y}}(8,8,2,1,1) + 2 \cdot \Sigma\bar{\mathbf{Y}}(8,8,1,1,1,1) + 17 \cdot \Sigma\bar{\mathbf{Y}}(8,7,5) + 18 \cdot \Sigma\bar{\mathbf{Y}}(8,7,4,1) + 6 \cdot \Sigma\bar{\mathbf{Y}}(8,7,3,2) + 20 \cdot \Sigma\bar{\mathbf{Y}}(8,7,3,1,1) + 10 \cdot \Sigma\bar{\mathbf{Y}}(8,7,2,2,1) + 18 \cdot \\ &\Sigma\bar{\mathbf{Y}}(8,7,2,1,1,1) + 14 \cdot \Sigma\bar{\mathbf{Y}}(8,7,1,1,1,1,1) + 18 \cdot \Sigma\bar{\mathbf{Y}}(8,6,6) + 6 \cdot \Sigma\bar{\mathbf{Y}}(8,6,5,1) + 13 \cdot \Sigma\bar{\mathbf{Y}}(8,6,4,2) + 5 \cdot \Sigma\bar{\mathbf{Y}}(8,6,4,1,1) + 1 \cdot \Sigma\bar{\mathbf{Y}}(8,6,3,3) + 17 \cdot \Sigma\bar{\mathbf{Y}}(8,6,3,2,1) + 3 \cdot \\ &\Sigma\bar{\mathbf{Y}}(8,6,3,1,1,1) + 20 \cdot \Sigma\bar{\mathbf{Y}}(8,6,2,2,2) + 13 \cdot \Sigma\bar{\mathbf{Y}}(8,6,2,2,1,1) + 5 \cdot \Sigma\bar{\mathbf{Y}}(8,6,2,1,1,1,1) + 9 \cdot \Sigma\bar{\mathbf{Y}}(8,6,1,1,1,1,1,1) + 5 \cdot \Sigma\bar{\mathbf{Y}}(7,7,6) + 1 \cdot \Sigma\bar{\mathbf{Y}}(7,7,5,1) + 6 \cdot \Sigma\bar{\mathbf{Y}}(7,7,4,2) + \\ &20 \cdot \Sigma\bar{\mathbf{Y}}(7,7,4,1,1) + 4 \cdot \Sigma\bar{\mathbf{Y}}(7,7,3,3) + 22 \cdot \Sigma\bar{\mathbf{Y}}(7,7,3,2,1) + 12 \cdot \Sigma\bar{\mathbf{Y}}(7,7,3,1,1,1) + 11 \cdot \Sigma\bar{\mathbf{Y}}(7,7,2,2,2) + 6 \cdot \Sigma\bar{\mathbf{Y}}(7,7,2,2,1,1) + 20 \cdot \Sigma\bar{\mathbf{Y}}(7,7,2,1,1,1,1) + 13 \cdot \\ &\Sigma\bar{\mathbf{Y}}(7,7,1,1,1,1,1,1) + 15 \cdot \Sigma\bar{\mathbf{Y}}(7,6,6,1) + 13 \cdot \Sigma\bar{\mathbf{Y}}(7,6,5,2) + 5 \cdot \Sigma\bar{\mathbf{Y}}(7,6,5,1,1) + 18 \cdot \Sigma\bar{\mathbf{Y}}(7,6,4,3) + 7 \cdot \Sigma\bar{\mathbf{Y}}(7,6,4,2,1) + 8 \cdot \Sigma\bar{\mathbf{Y}}(7,6,4,1,1,1) + 20 \cdot \Sigma\bar{\mathbf{Y}}(7,6,3,3,1) + 10 \cdot \\ &\Sigma\bar{\mathbf{Y}}(7,6,3,2,2) + 18 \cdot \Sigma\bar{\mathbf{Y}}(7,6,3,2,1,1) + 14 \cdot \Sigma\bar{\mathbf{Y}}(7,6,3,1,1,1,1) + 9 \cdot \Sigma\bar{\mathbf{Y}}(7,6,2,2,2,1) + 7 \cdot \Sigma\bar{\mathbf{Y}}(7,6,2,2,1,1,1) + 8 \cdot \Sigma\bar{\mathbf{Y}}(7,6,2,1,1,1,1,1) + 19 \cdot \Sigma\bar{\mathbf{Y}}(7,6,1,1,1,1,1,1,1) \\ &\mod 23 \end{aligned}$$

We checked every rational $\eta = d/k$ such that $k \leq 20$, and verified that our framework does in fact yield a 2-party PSM protocol with unbalanced communication complexity $O(N^{\eta})$, $O(N^{1-\eta})$.

[4] The source code can be downloaded from https://github.com/tianren/psm.

5 Open Problems

This paper presents two frameworks: a framework of constructing k-party PSM protocols for general $f : [N]^k \to \{0,1\}$ with c.c. $O_k(N^{\frac{k-1}{2}})$, and a framework of constructing 2-party PSM protocols for general $f : [N] \times [N] \to \{0,1\}$ where one party sends $O_\eta(N^\eta)$ bits and the other party sends $O_\eta(N^{1-\eta})$ bits. An immediate open problem is to prove our frameworks work for all integer k and all rational η. Currently, we can only prove it works for some k and η.

For simplicity, our analysis only considers the symmetric sum of terms. The symmetric sum incurs a blow-up exponential on k. Thus the communication complexity of our k-party PSM protocols is $\exp(k) \cdot N^{\frac{k-1}{2}}$. While [BKN18] achieves communication complexity $\text{poly}(k) \cdot N^{\frac{k}{2}}$. Our protocols are less efficient in the domain where $\log N < k$. A possible approach of getting rid of the exponential dependency in k is to break the symmetry. The potential of such an approach is evidenced by the 5-party PSM protocol in Section 1.2, which is asymmetric.

There is no clear reason why our framework will not yields more efficient PSM protocols. Can our multi-party framework yield PSM protocols with communication complexity $O_k(N^{\frac{k}{2}-1})$, when k is sufficiently large? Can our 2-party framework might yield PSM protocol with communication $O_\eta(N^\eta)$ for some rational $\eta < \frac{1}{2}$? Our technique transfers such questions into some linear systems. Each question has an affirmative answer (for a given k or η) if and only if the corresponding linear system is solvable. We have modified our program to generate and solve these linear systems, but all the system we have tried are unsolvable. The failure suggests that our new upper bounds might be tight, or are tight for a natural class of PSM protocols.

The question of the communication complexity trade-off for multi-party PSM remains widely open. In our k-party PSM protocol, every party sends $O_k(N^{\frac{k-1}{2}})$ bits. A variant of [FKN94] provides a k-party PSM protocol where the i-th party sends $\tilde{O}_k(N^{i-1})$ bits, whose geometric average is $\tilde{O}_k(N^{\frac{k-1}{2}})$. Should a future work achieves the smooth trade-off between the two, there is little doubt that it will bring us a deep insight into PSM.

Finally, this work belongs to a not-fully-successful attempt at constructing PSM with sub-exponential communication complexity, which is probably the moonshot open problem in the PSM literature.

Acknowledgement

We would like to thank Hoeteck Wee, Vinod Vaikuntanathan amd Michel Abdalla for helpful discussions. TL was supported by NSF grants CNS-1528178, CNS-1929901, CNS-1936825 (CAREER), CNS-2026774, a JP Morgan AI research Award, and a Simons Foundation Collaboration Grant on Algorithmic Fairness. Part of this work was performed while TL was in MIT, during which he was supported in part by NSF Grants CNS-1350619, CNS-1414119 and CNS-1718161, an MIT-IBM grant and a DARPA Young Faculty Award. LA was supported by a doctoral grant from the French Ministère de l'Enseignement Supérieur et de la Recherche.

References

[AA18] Benny Applebaum and Barak Arkis. On the power of amortization in secret sharing: d-uniform secret sharing and CDS with constant information rate. In Amos Beimel and Stefan Dziembowski, editors, *Theory of Cryptography - 16th International Conference, TCC 2018, Panaji, India, November 11-14, 2018, Proceedings, Part I*, volume 11239 of *Lecture Notes in Computer Science*, pages 317–344. Springer, 2018.

[AARV17] Benny Applebaum, Barak Arkis, Pavel Raykov, and Prashant Nalini Vasudevan. Conditional disclosure of secrets: Amplification, closure, amortization, lower-bounds, and separations. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:38, 2017.

[ABF+19] Benny Applebaum, Amos Beimel, Oriol Farràs, Oded Nir, and Naty Peter. Secret-sharing schemes for general and uniform access structures. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 441–471. Springer, 2019.

[AHMS20] Benny Applebaum, Thomas Holenstein, Manoj Mishra, and Ofer Shayevitz. The communication complexity of private simultaneous messages, revisited. *J. Cryptol.*, 33(3):917–953, 2020.

[BHI+20] Marshall Ball, Justin Holmgren, Yuval Ishai, Tianren Liu, and Tal Malkin. On the complexity of decomposable randomized encodings, or: How friendly can a garbling-friendly PRF be? In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPIcs*, pages 86:1–86:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[BIK17] Amos Beimel, Yuval Ishai, and Eyal Kushilevitz. Ad hoc PSM protocols: Secure computation without coordination. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III*, volume 10212 of *Lecture Notes in Computer Science*, pages 580–608, 2017.

[BIKK14] Amos Beimel, Yuval Ishai, Ranjit Kumaresan, and Eyal Kushilevitz. On the cryptographic complexity of the worst functions. In *TCC*, pages 317–342, 2014.

[BKN18] Amos Beimel, Eyal Kushilevitz, and Pnina Nissim. The complexity of multiparty PSM protocols and related models. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 287–318. Springer, 2018.

[CGO21] Michele Ciampi, Vipul Goyal, and Rafail Ostrovsky. Threshold garbled circuits and ad hoc secure computation. Cryptology ePrint Archive, Report 2021/308, 2021. https://eprint.iacr.org/2021/308.

[FKN94] Uriel Feige, Joe Kilian, and Moni Naor. A minimal model for secure computation (extended abstract). In Frank Thomson Leighton and Michael T. Goodrich, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 554–563. ACM, 1994.

[GKW15] Romain Gay, Iordanis Kerenidis, and Hoeteck Wee. Communication complexity of conditional disclosure of secrets and attribute-based encryption. In *CRYPTO (II)*, pages 485–502, 2015.

[IK97] Yuval Ishai and Eyal Kushilevitz. Private simultaneous messages protocols with applications. In *Fifth Israel Symposium on Theory of Computing and Systems, ISTCS 1997, Ramat-Gan, Israel, June 17-19, 1997, Proceedings*, pages 174–184. IEEE Computer Society, 1997.

[IK00] Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 294–304. IEEE Computer Society, 2000.

[LVW17] Tianren Liu, Vinod Vaikuntanathan, and Hoeteck Wee. Conditional disclosure of secrets via non-linear reconstruction. In *CRYPTO Part I*, pages 758–790, 2017.

A Proof of Equation (9) and (10)

Proof of Equation (9). By definition:

$$\sum\langle \mathbf{F}, \bar{\mathbf{X}}(P)\rangle = \sum_{(*)} \langle \mathbf{F}, \bar{\mathbf{X}}_{S_1} \otimes \ldots \otimes \bar{\mathbf{X}}_{S_t}\rangle$$

where $(*)$ denotes "for all unordered $E = \{S_1, \ldots, S_t\}$ being a partition of $[2k]$ such that $\{|S_1|, \ldots, |S_t|\} = P$". Thus,

$$\begin{aligned}
\sum\langle \mathbf{F}, \bar{\mathbf{X}}(P)\rangle &= \sum_{(*)} \Big\langle \mathbf{F}, \bigotimes_{i\in[t]} (\mathbf{R}_{S_i} + \bigotimes_{j\in S_i} \mathbf{x}_j)\Big\rangle \\
&= \sum_{(*)} \sum_{G\subseteq E} \Big\langle \mathbf{F}, \bigotimes_{S\in G} \mathbf{R}_S \otimes \bigotimes_{j\notin \bigcup_{S\in G} S} \mathbf{x}_j \Big\rangle \\
&= \sum_{Q\subseteq P} \sum_{\substack{G=\{S_1,\ldots,S_t\} \text{ s.t.} \\ \{|S_1|,\ldots,|S_t|\}=P\setminus Q}} \beta(P,G) \cdot \Big\langle \mathbf{F}, \bigotimes_{i\in[t]} R_{S_i} \otimes \bigotimes_{j\notin \bigcup_{i\in[t]} S_i} \mathbf{x}_j \Big\rangle,
\end{aligned}$$

where $\beta(P,G)$ accounts for the redundancy: define $\beta(P,G)$ as the number of unordered partitions E of $[2k]$ such that $G \subseteq E$ and P is the shape of E. It is equivalent to count the number of $F := E \setminus G$. That is, $\beta(P,G)$ also equals the number of unordered partitions F of $[2k] \setminus \bigcup_{S\in G} S$ such that Q is the shape of F. Thus by definition, $\beta(P,G) = \alpha(Q)$. The proof is concluded by

$$\begin{aligned}
\sum\langle \mathbf{F}, \bar{\mathbf{X}}(P)\rangle &= \sum_{Q\subseteq P} \sum_{\substack{G=\{S_1,\ldots,S_t\} \text{ s.t.} \\ \{|S_1|,\ldots,|S_t|\}=P\setminus Q}} \alpha(Q) \cdot \Big\langle \mathbf{F}, \bigotimes_{i\in[t]} R_{S_i} \otimes \bigotimes_{j\notin \bigcup_{i\in[t]} S_i} \mathbf{x}_j \Big\rangle \\
&= \sum_{Q\subseteq P} \alpha(Q) \cdot \sum\langle \mathbf{F}, \mathbf{R}(P\setminus Q)\rangle. \qquad \square
\end{aligned}$$

Proof of Equation (10). Let $n = \mathrm{sum}(Q)$. By definition, $\alpha(Q)$ is the number of unsorted partitions $E = \{S_1, \ldots, S_t\}$ of $[n]$ such that the multiset $\{|S_1|, \ldots, |S_t|\}$ (i.e. the shape of E) equals Q.

To compute $\alpha(Q)$, we count the number of ways to arranging $1, \ldots, n$ into a sequence.

- First, pick an unsorted partitions E of $[n]$ s.t. the shape of E equals Q. The number of choices is $\alpha(Q)$.

- Then, sort the sets in the partion $E = \{S_1, \ldots, S_t\}$. Sort them by their sizes, i.e. $|S_1| \leq |S_2| \leq \cdots \leq |S_t|$. For any m, if several sets are of the size m, their order has to be specified, the number of such choices is (number of m's in Q)!.
- Finally, arrange the elements in each S_i into a sub-sequence, the number of possible sequences is $|S_i|!$. Concatenate these sub-sequences in order.

$$\alpha(Q) \cdot \prod_{m \in \mathbb{Z}^+} (\text{number of } m\text{'s in } Q)! \cdot \prod_{i \in Q} i! = n! \qquad \square$$

B Auxiliary PSM Protocols for $\langle \mathbf{x}_1 \otimes \ldots \otimes \mathbf{x}_k, \mathbf{Y} \rangle + s$

B.1 The Multi-party Variant

In this section, we present an auxiliary PSM protocol that is used as a subroutine by our multi-party PSM in Section 3.

The functionality is $\langle \mathbf{x}_1 \otimes \ldots \otimes \mathbf{x}_k, \mathbf{Y} \rangle + s$. It is a $(k+1)$-party functionality where the i-th party has as input $\mathbf{x}_i \in \mathbb{F}^N$ for $i \in [k]$, and the $(k+1)$-th party has as inputs $\mathbf{Y} \in \mathbb{F}^{N \times \cdots \times N}_{k \text{ times}}$ and $s \in \mathbb{F}$. We will present a PSM protocol for this functionality with a communication complexity of $O(\text{poly}(k) \cdot N^k)$ field elements. This protocol is implicitly used in [BKN18].

First, we consider the special case when $k = 1$. That is, there are only two parties. Say we call them Alice and Bob. Alice has $\mathbf{x} \in \mathbb{F}^N$, Bob has $\mathbf{y} \in \mathbb{F}^N, s \in \mathbb{F}$. The functionality output is $\langle \mathbf{x}, \mathbf{y} \rangle + s$. The PSM protocol works as follows:

- Random $\mathbf{a}, \mathbf{b} \in \mathbb{F}^N, c \in \mathbb{F}$ are sampled from the common random string, which is known by both Alice and Bob.
- Alice sends $\bar{\mathbf{x}} := \mathbf{x} + \mathbf{a}, z := c - \langle \mathbf{b}, \mathbf{x} \rangle$ to the referee.
- Bob sends $\bar{\mathbf{y}} := \mathbf{y} + \mathbf{b}, w := s - c - \langle \mathbf{a}, \mathbf{y} \rangle - \langle \mathbf{a}, \mathbf{b} \rangle$ to the referee.
- The referee outputs $\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle + z + w$.

For the case $k \geq 2$, the first k parties need to jointly emulate Alice. The protocol works as follows:

- Random $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{F}^{N \times \cdots \times N}$ are sampled from the common random string. Define $c \in \mathbb{F}$ as the sum of entries in $\mathbf{C}$.
- The $(k+1)$-th party sends $\bar{\mathbf{Y}} := \mathbf{Y} + \mathbf{B}, z := s - c - \langle \mathbf{A}, \mathbf{Y} \rangle - \langle \mathbf{A}, \mathbf{B} \rangle$ to the referee.
- The first k parties jointly reveal $\bar{\mathbf{X}} := \mathbf{x}_1 \otimes \ldots \otimes \mathbf{x}_k + \mathbf{A}$, $w := c - \langle \mathbf{B}, \mathbf{x}_1 \otimes \ldots \otimes \mathbf{x}_k \rangle$ to the referee.

  Since every coordinate of $\bar{\mathbf{X}}$ can be computed by an arithmetic formula of size $O(k)$, each of these coordinates can be computed by the referee by using a PSM protocol with communication complexity of $O(\text{poly}(k))$ field elements [IK00]. The referee learns $\bar{\mathbf{X}}$ after receiving $O(\text{poly}(k) \cdot N^k)$ field elements.

  The term $w := c - \langle \mathbf{B}, \mathbf{x}_1 \otimes \ldots \otimes \mathbf{x}_k \rangle$ equals the sum of all entries in $\mathbf{W} := \mathbf{C} - \mathbf{B} \circ_{\text{p.w.}} (\mathbf{x}_1 \otimes \ldots \otimes \mathbf{x}_k)$, where $\circ_{\text{p.w.}}$ denotes the point-wise product. In other words, we defines $\mathbf{W} \in \mathbb{F}^{N \times \cdots \times N}$ as

$$\mathbf{W}[i_1, \ldots, i_k] = \mathbf{C}[i_1, \ldots, i_k] - \mathbf{B}[i_1, \ldots, i_k] \mathbf{x}_1[i_1] \ldots \mathbf{x}_k[i_k].$$

Due to the randomness of $\mathbf{C}$, we know $\mathbf{W}$ is a randomized encoding of w. Thus, it is equivalent for the first k parties to jointly reveal $\mathbf{W}$ to the referee. Since every coordinate of $\mathbf{W}$ can be computed by an arithmetic formula of size $O(k)$, each of them can be revealed by using the Ishai-Kushilevitz PSM protocol [IK00], which has a communication complexity of $O(\text{poly}(k))$ field elements. The referee learns w after receiving $O(\text{poly}(k) \cdot N^k)$ field elements.

- The referee outputs $\langle \bar{\mathbf{X}}, \bar{\mathbf{Y}} \rangle + z + w$.

The correctness of the protocol can be verified in the following equation:

$$\begin{aligned} & \langle \bar{\mathbf{X}}, \bar{\mathbf{Y}} \rangle + z + w \\ = & \langle \mathbf{x}_1 \otimes \ldots \otimes \mathbf{x}_k + \mathbf{A}, \mathbf{Y} + \mathbf{B} \rangle + s - c - \langle \mathbf{A}, \mathbf{Y} \rangle - \langle \mathbf{A}, \mathbf{B} \rangle + \\ & c - \langle \mathbf{B}, \mathbf{x}_1 \otimes \ldots \otimes \mathbf{x}_k \rangle \\ = & \langle \mathbf{x}_1 \otimes \ldots \otimes \mathbf{x}_k, \mathbf{Y} \rangle + s. \end{aligned}$$

The privacy is guaranteed by the following simulator:

- Simulate $\bar{\mathbf{X}}, \bar{\mathbf{Y}}, \mathbf{W}$ as uniform random, since they are one-time-padded by $\mathbf{A}, \mathbf{B}, \mathbf{C}$.
- Given $\bar{\mathbf{X}}, \bar{\mathbf{Y}}, \mathbf{W}$ and the function output, w, z are uniquely determined since $w = \sum(\mathbf{W})$ and $\langle \bar{\mathbf{X}}, \bar{\mathbf{Y}} \rangle + z + w = \text{output}$.
- Simulate the transcripts of the inner Ishai-Kushilevitz PSM protocols using its own simulator, which takes $\bar{\mathbf{X}}, \mathbf{W}$ as input.

B.2 The 2-party Variant

In this section, we present an auxiliary PSM protocol that is used as a subroutine by our unbalanced 2-party PSM in Section 4.

The functionality is $\langle \mathbf{x}_1 \otimes \ldots \otimes \mathbf{x}_k, \mathbf{Y} \rangle + s$. It is a 2-party functionality where the first party, namely Alice, has as inputs $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbb{F}^N$ and the second party, namely Bob, has as inputs $\mathbf{Y} \in \mathbb{F}^{N \times \cdots \times N}_{k \text{ times}}$ and $s \in \mathbb{F}$. We will present a PSM protocol for this functionality with unbalanced communication complexity, where Alice sends $O(kN)$ field elements and Bob sends $(N+1)^k$ field elements.

As the first step, we consider a harder problem instead. Bob's input is replaced by a multi-affine function $f : \mathbb{F}^N \times \cdots \times \mathbb{F}^N \to \mathbb{F}$. Corresponding, the functionality is replaced by $f(\mathbf{x}_1, \ldots, \mathbf{x}_k)$. Every multi-affine function f can be uniquely represented by its coefficient tensor $\mathbf{F} \in \mathbb{F}^{(N+1) \times \cdots \times (N+1)}$ such that for any $\mathbf{z}_1, \ldots, \mathbf{z}_k \in \mathbb{F}^N$,

$$f(\mathbf{z}_1, \ldots, \mathbf{z}_k) = \langle \mathbf{z}_1 \| 1 \otimes \cdots \otimes \mathbf{z}_k \| 1, \mathbf{F} \rangle.$$

Here $\mathbf{z}_i \| 1$ denotes the concatenation of $\mathbf{z}_i$ and 1, which is a dimension-$(N+1)$ vector. Notice that, if we let the "first" $N \times \cdots \times N$ subtensor of $\mathbf{F}$ equal $\mathbf{Y}$, let its "last" entry $\mathbf{F}[N+1, \ldots, N+1] = s$, and let all other entries in $\mathbf{F}$ be 0, we have

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \langle \mathbf{x}_1 \| 1 \otimes \cdots \otimes \mathbf{x}_k \| 1, \mathbf{F} \rangle = \langle \mathbf{x}_1 \otimes \ldots \otimes \mathbf{x}_k, \mathbf{Y} \rangle + s.$$

The protocol works as follows:

- Random $\mathbf{r}_1, \ldots, \mathbf{r}_k \in \mathbb{F}^N$ and a random multi-affine function g are sampled from the common random string.

- Alice sends $\bar{\mathbf{x}}_i = \mathbf{x}_i + \mathbf{r}_i$ to the referee, for all $i \in [k]$.
- Bob computes the multi-affine function g, such that

$$g(\mathbf{z}_1, \ldots, \mathbf{z}_k) := f(\mathbf{z}_1 - \mathbf{r}_1, \ldots, \mathbf{z}_k - \mathbf{r}_k).$$

  Bob sends $\bar{g} = g + h$ to the referee.
- Alice additionally sends $s = h(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_k)$ to the referee.
- The referee outputs $\bar{g}(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_k) - s$.

The correctness follows directly from the following equation:

$$\begin{aligned}
\bar{g}(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_k) - s &= g(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_k) + h(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_k) - h(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_k) \\
&= g(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_k) \\
&= f(\mathbf{x}_1 - \mathbf{r}_1 + \mathbf{r}_1, \ldots, \mathbf{x}_k - \mathbf{r}_k + \mathbf{r}_k) \\
&= f(\mathbf{x}_1, \ldots, \mathbf{x}_k).
\end{aligned}$$

The privacy is guaranteed by the following simulator:

- Simulate $\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_k, \bar{g}$ as uniform random, since they are one-time padded by $\mathbf{r}_1, \ldots, \mathbf{r}_k, h$.
- Given $\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_k, \bar{g}$ and the function output, simulate s by computing s from the equation $\bar{g}(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_k) - s = \text{output}$.