

# Distributed Private Data Analysis: On Simultaneously Solving *How* and *What*\*

Amos Beimel   Kobbi Nissim   Eran Omri

Department of Computer Science

Ben Gurion University

Be'er Sheva, Israel

{beimel, kobbi, omri}@cs.bgu.ac.il

May 2, 2010

## Abstract

We examine the combination of two directions in the field of privacy concerning computations over distributed private inputs – *secure function evaluation* (SFE) and *differential privacy*. While in both the goal is to privately evaluate some function of the individual inputs, the privacy requirements are significantly different. The general feasibility results for SFE suggest a natural paradigm for implementing differentially private analyses distributively: First choose *what* to compute, i.e., a differentially private analysis; Then decide *how* to compute it, i.e., construct an SFE protocol for this analysis.

We initiate an examination whether there are advantages to a paradigm where both decisions are made simultaneously. In particular, we investigate under which accuracy requirements it is beneficial to adapt this paradigm for computing a collection of functions including binary sum, gap threshold, and approximate median queries. Our results imply that when computing the binary sum of  $n$  distributed inputs then:

- When we require that the error is  $o(\sqrt{n})$  and the number of rounds is constant, there is no benefit in the new paradigm.
- When we allow an error of  $O(\sqrt{n})$ , the new paradigm yields more efficient protocols when we consider protocols that compute symmetric functions.

Our results also yield new separations between the local and global models of computations for private data analysis.

**Keywords.** Differential privacy, Secure Function Evaluation, Sum Queries.

---

\*A preliminary version of this work appeared in David Wagner editor, *Advances in Cryptology – CRYPTO 2008*. Volume 5157 of *Lecture Notes in Computer Science*, pages 451–468. Springer, 2008.

# 1 Introduction

We examine the combination of two directions in the field of privacy concerning distributed private inputs – secure function evaluation [26, 18, 4, 1] and differential privacy [11, 8]. While in both the goal is to privately evaluate some function of individual inputs, the privacy requirements are significantly different.

Secure function evaluation (SFE) allows  $n$  parties  $p_1, \dots, p_n$ , sharing a common interest in distributively computing a function  $f(\cdot)$  of their inputs  $\mathbf{x} = (x_1, \dots, x_n)$ , to compute  $f(\mathbf{x})$  while making sure that no coalition of  $t$  or less curious parties learns more than the outcome of  $f(\mathbf{x})$ . I.e., for every such coalition, executing the SFE protocol is equivalent to communicating with a trusted party that is given the private inputs  $\mathbf{x}$  and releases  $f(\mathbf{x})$ . SFE has been the subject of extensive cryptographic research (initiated in [26, 18, 4, 1]), and SFE protocols exist for any feasible function  $f(\cdot)$  in a variety of general settings.

SFE is an important tool for achieving privacy of individual entries – no information about these entries is leaked beyond the outcome  $f(\mathbf{x})$ . However this guarantee is insufficient in many applications, and care must be taken in choosing the function  $f(\cdot)$  to be computed – any implementation, no matter how secure, of a function  $f(\cdot)$  that leaks individual information would not preserve individual privacy.

A criterion for functions that preserve privacy of individual entries, *differential privacy*, has evolved in a sequence of recent works [7, 15, 14, 2, 11, 8, 9]. It has been demonstrated that differentially private analyses exist for a variety of tasks including the approximation of numerical functions (by adding carefully chosen random noise that conceals any single individual’s contribution) [11, 2, 22, 17], non-numerical analyses [20], datamining [2, 22], learning [2, 19], non-interactive sanitization [3, 13, 16], and statistical analysis [10, 24].

Employing the generality of SFE, we can combine these two directions in a natural paradigm for constructing protocols in which differential privacy is preserved:

1. Decide on *what* to compute. This can be, e.g., a differentially private analysis  $\hat{f}(\cdot)$  that approximates a desired analysis  $f(\cdot)$ . Designing  $\hat{f}(\cdot)$  can be done while abstracting out all implementation issues, assuming the computation is performed by a trusted party that only announces the outcome of the analysis.
2. Decide on *how* to compute, i.e., construct an SFE protocol for computing  $\hat{f}(\mathbf{x})$  either by using one of the generic transformations of the feasibility results mentioned above, or by crafting an efficient protocol that utilizes the properties of  $\hat{f}(\cdot)$ .

This natural paradigm yields a conceptually simple recipe for constructing distributed analyses preserving differential privacy, and, furthermore, allows a valuable separation of our examinations of the *what* and *how* questions.

Comparing the privacy requirements of SFE protocols with differential privacy suggests, however, that this combination may result in sub-optimal protocols. For example, differential privacy is only concerned with how the view of a coalition changes when one (or only few) of the inputs are changed, whereas SFE protocols are required to keep these views indistinguishable even when significant changes occur, if these changes do not affect the computed function’s outcome. Hence, it may be advantageous to consider a paradigm where the analysis to be computed and the protocol for computing it are chosen simultaneously.

## 1.1 Our Underlying Models

The main distributed model we consider is of  $n$  honest-but-curious (a.k.a. semi-honest) parties  $p_1, \dots, p_n$  that are connected via a complete network and perform a computation over their private inputs  $x_1, \dots, x_n$ .

Privacy is required to be maintained with respect to all coalitions of size up to  $t$ . The model of honest-but-curious parties has been examined thoroughly in cryptography, and was shown to enable SFE in a variety of settings [26, 18, 1, 4]. We change the standard definition so that differential privacy has to be maintained with respect to coalitions of curious parties (see Definition 2.4 below).

Another distributed model we consider is the *local model*<sup>1</sup>. Protocols executing in the local model have a very simple communication structure, where each party  $p_i$  can only communicate with a designated honest-but-curious party  $C$ , which we refer to as a *curator*. The communication can either be *non-interactive*, where each party sends a single message to the curator which replies with the protocol’s outcome, or *interactive*, where several rounds of communication may take place.

While it is probably most natural to consider a setting where the players are computationally limited (i.e., all are probabilistic polynomial time machines), we present our results in an information theoretic setting. This choice has two benefits:

- Technically, it allows us to prove lower bounds on SFE protocols (where similar bounds are not known for the computational setting). Hence, we can rigorously demonstrate when constructing differentially private protocols is better than using the natural paradigm.
- On the flip side, our bounds on the information theoretic model demonstrate, for the first time, a setting where reliance on computational hardness assumptions strictly improves the construction of differentially private analyses.

## 1.2 Our Results

We initiate an examination of the paradigm where an analysis and the protocol for computing it are chosen simultaneously. We begin with two examples that present the potential benefits of using this paradigm: it can lead to simpler protocols, and more importantly it can lead to more efficient protocols. For the latter we consider the Binary Sum function,

$$\text{SUM}(x_1, \dots, x_n) = \sum_{i=1}^n x_i \quad \text{for } x_i \in \{0, 1\}.$$

The major part of this work examines whether constructing non-SFE protocols for computing an approximation  $\hat{f}(\cdot)$  to  $\text{SUM}(\cdot)$  yields an efficiency gain<sup>2</sup>. Ignoring the dependency on the privacy parameter, our first observation is that for approximations with additive error  $\approx \sqrt{n}$  there is a gain – for a natural class of *symmetric* approximation functions (informally, functions where the outcome does not depend on the order of inputs), it is possible to construct differentially private protocols that are much more efficient than any SFE protocol for a function in this class. Moreover, these differentially private protocols are secure against coalitions of size up to  $t = n - 1$ , and need not rely on secure channels.

The picture changes when we consider additive error smaller than  $\sqrt{n}$ . This follows from a sequence of results:

1. We prove first that no such non-interactive protocols in the local model exist. Furthermore, no local protocols with  $\ell \leq \sqrt{n}$  rounds and additive error  $\sqrt{n}/\tilde{O}(\ell)$  exist.

<sup>1</sup>Also referred to in the literature as *randomized response* and *input perturbation*. This model was originally introduced by Warner [25] as a means of encouraging survey responders to answer truthfully, and has been studied extensively since.

<sup>2</sup>We only consider *oblivious protocols* where the communication pattern is independent of input and randomness (see Section 2.2).

2. We show that in particular, no local interactive protocol with  $o(\sqrt{n/\log n})$  rounds exists for computing  $\text{SUM}(\cdot)$  within constant additive error (this is in contrast to the centralized setup where  $\text{SUM}(\cdot)$  can be computed within  $O(1)$  additive error).
3. Finally, we prove that the bounds on local protocols imply that no distributed protocols exist that use  $nt/4$  messages, and approximates  $\text{SUM}(\cdot)$  within additive error  $\sqrt{n}/\tilde{O}(\ell)$  in  $\ell$  rounds.

Considering the natural paradigm, i.e., computing a differentially-private approximation to  $\text{SUM}(\cdot)$  using SFE, we get a protocol for approximating  $\text{SUM}(\cdot)$  with  $O(1)$  additive error, and sending  $O(nt)$  messages. Thus, for protocols with error  $o(\sqrt{n}/\varepsilon)$  and small number of rounds, there is no gain in using the paradigm of a simultaneous design of the function and its protocol.

Our results imply that differentially private protocols constructed under computational hardness assumptions, yielding a computational version of differential privacy (see Definition 2.5), are provably more efficient than protocols that do not make use of computational hardness. For instance, the phase transition we observe at  $\theta(\sqrt{n}/\varepsilon)$  additive error *does not* hold in a computational setting. See Example 2.6 for details.

### 1.3 Techniques

We prove our lowerbound for the distributed model in a sequence of reductions. We begin with a simple reduction from any differentially private protocol for  $\text{SUM}$  to a gap version of the threshold function, denoted GAP-TR. Henceforth, it is enough to prove our lowerbound for GAP-TR.

In the heart of our lowerbound for GAP-TR is a transformation from efficient distributed protocols into local interactive protocols, showing that if there are distributed differentially-private protocols for GAP-TR( $\cdot$ ) in which half of the parties interact with less than  $t + 1$  parties, then there exist differentially-private protocols for GAP-TR( $\cdot$ ) in the local interactive model. This allows us to prove our impossibility results in the local model, which is considerably simpler to analyze.

In analyzing the local non-interactive model, we prove lowerbounds borrowing from analyses in [7, 14]. The main technical difference is that our analysis is a lowerbound and hence should hold for general protocols, whereas the work in [7, 14] was concerned with proving feasibility of privacy preserving computations (i.e., upperbounds), and hence they analyze of very specific protocols.

To extend our lowerbounds from the local non-interactive to interactive protocols, we decompose an  $\ell$ -round interactive protocol to  $\ell$  one-round protocols, analyze the  $\ell$  protocols, and use composition to obtain the lowerbound.

### 1.4 Related Work

Secure function evaluation and private data analysis were first tied together in the *Our Data, Ourselves (ODO)* protocols [9]. The constructions in [9] – distributed SFE protocols for generating shares of random noise used in private data analyses – follow the natural paradigm discussed above (however, they avoid utilizing generic SFE feasibility results to gain on efficiency). We note that a difference between the protocols in [9] and the discussion herein is that ODO protocols are secure against malicious parties, in a computational setup, whereas we deal with honest-but-curious parties, and mostly in an information theoretic setup. Following our work, computational differential privacy was considered in [21]; they present several definitions of computational differential privacy, study the relationships between these definitions, and construct efficient 2-party computational differentially private protocols for approximating the distance between two vectors. In this work, we supply a definition of computationally  $(t, \epsilon)$ -differentially private protocols which is close to the definition of IND-CDP privacy in [21].

Lowerbounds on the local non-interactive model were previously presented implicitly in [11, 23, 19], and explicitly in [7, 12]. The two latter works are mainly concerned with what is called the global (or centralized) interactive setup, but have also implications to approximation to SUM in the local *non-interactive* model, namely, that it is impossible to approximate it within additive error  $c\sqrt{n}$  (for some constant  $c > 0$ ), a slightly weaker result compared to our lowerbound of  $c\sqrt{n}/\varepsilon$  for  $\varepsilon$ -differentially private local non-interactive protocols. However, (to the best of our understanding) these implications of [7, 12] do not imply the lowerbounds we get for local interactive protocols and distributed protocols.

Chor and Kushilevitz [5] consider the problem of securely computing modular sum when the inputs are distributed. They show that this task can be done while sending roughly  $n(t+1)/2$  messages. Furthermore, they prove that this number of messages is optimal for a family of protocols that they call oblivious. These are protocols where the communication pattern is fixed and does not depend on the inputs or random inputs. In our work we extend their lowerbound result and prove that with  $n(t+1)/4$  messages no symmetric approximation for SUM with sub-linear additive error can be computed in an oblivious protocol.

## 1.5 Organization

The rest of the paper is organized as follows: In Section 2 we define differentially private analyses and its extension to differentially private protocols (both information-theoretic and computational), describe the local model of communication, and define the binary sum and gap threshold functions. In Section 3, we present two motivating examples for our new methodology of simultaneously solving how and what. In Section 4 we prove lowerbounds on the error of differentially private protocols for computing the binary sum and gap threshold functions in the local model, and in Section 5 we extend these lowerbounds to the distributed model. Finally, in Section 6 we prove that an SFE protocol for computing a symmetric approximation of the sum function with less than  $nt/4$  messages has an error of  $\Omega(n)$  (compared to a non-SFE protocol that approximates the sum function with  $O(n)$  messages and an error of  $\Omega(\sqrt{n})$ ).

## 2 Preliminaries

**Notation.** A vector  $\mathbf{x} = (x_1, \dots, x_n)$  is an ordered sequence of  $n$  elements of some domain  $D$ . Vectors  $\mathbf{x}, \mathbf{x}'$  are *neighboring* if they differ on exactly one entry, and are *T-neighboring* if they differ on a single entry whose index is *not* in  $T \subset [n]$ .

The *Laplace distribution*,  $\text{Lap}(\lambda)$ , is the continuous probability distribution with probability density function

$$h(y) = \frac{\exp(-|y|/\lambda)}{2\lambda}.$$

For  $Y \sim \text{Lap}(\lambda)$  we have that  $\mathbb{E}[Y] = 0$ ,  $\text{Var}[Y] = 2\lambda^2$ , and  $\Pr[|Y| > k\lambda] = e^{-k}$ .

**Definition 2.1** Let  $D_I, D_R$ , and  $R$  be sets. An  $n$ -ary randomized function is a function  $\hat{f} : (D_I)^n \times D_R \rightarrow R$ , where  $D = D_I$  is the domain of  $\hat{f}$  and  $D_R$  is the set of random inputs. For  $\mathbf{x} = (x_1, \dots, x_n) \in D^n$  we usually write  $\hat{f}(\mathbf{x})$  with the underlying convention that  $\hat{f}(x_1, \dots, x_n) = \hat{f}(x_1, \dots, x_n, r)$ , where  $r$  is uniformly selected from  $D_R$ . Following this convention, we also usually omit  $D_R$  from the notation and write  $\hat{f} : D^n \rightarrow R$ .

## 2.1 Differential Privacy

Our privacy definition for distributed protocols (Definition 2.4 below) can be viewed as a distributed variant of  $\varepsilon$ -differential privacy. Informally, a computation is differentially private if any change in a single individual input may only induce a small change in the distribution on its outcomes.

**Definition 2.2 (Differential privacy [11])** *Let  $\hat{f} : \mathcal{D}^n \rightarrow R$  be a randomized function from domain  $\mathcal{D}^n$  to range  $R$ . We say that  $\hat{f}$  is  $\varepsilon$ -differentially private if for all neighboring vectors  $\mathbf{x}, \mathbf{x}'$ , and for all possible sets of outcomes  $\mathcal{V} \subseteq R$  it holds that*

$$\Pr[\hat{f}(\mathbf{x}) \in \mathcal{V}] \leq e^\varepsilon \cdot \Pr[\hat{f}(\mathbf{x}') \in \mathcal{V}].$$

*The probability is taken over the randomness of  $\hat{f}$ .*

Several frameworks for constructing differentially private functions by means of perturbation are presented in the literature (see [11, 2, 22, 20]). The most basic transformation on a function  $f$  that yields a differentially private function is via the framework of *global sensitivity* [11]. In this framework the outcome is obtained by adding to  $f(\mathbf{x})$  noise sampled from the Laplace distribution, calibrated to the global sensitivity of  $f$ , defined as

$$\text{GS}_f = \max |f(\mathbf{x}) - f(\mathbf{x}')|, \text{ with the maximum taken over neighboring } \mathbf{x}, \mathbf{x}'.$$

Formally,  $\hat{f}$  is defined as

$$\hat{f}(\mathbf{x}) = f(\mathbf{x}) + Y, \text{ where } Y \sim \text{Lap}(\text{GS}_f/\varepsilon). \quad (1)$$

**Example 2.3** *The binary sum function  $\text{SUM} : \{0, 1\}^n \rightarrow \mathbb{R}$  is defined as  $\text{SUM}(\mathbf{x}) = \sum_{i=1}^n x_i$ . For every two neighboring  $\mathbf{x}, \mathbf{x}' \in \{0, 1\}^n$  we have that  $|\text{SUM}(\mathbf{x}) - \text{SUM}(\mathbf{x}')| = 1$  and hence  $\text{GS}_{\text{SUM}} = 1$ . Applying Equation (1), we get an  $\varepsilon$ -differentially private approximation,  $\hat{f}(\mathbf{x}) = \text{SUM}(\mathbf{x}) + Y$ , where  $Y \sim \text{Lap}(1/\varepsilon)$ , that is, we get a differentially private approximation of  $\text{SUM}$  with  $O(1)$  additive error.*

## 2.2 Differentially Private Protocols

We consider a distributed setting, where  $n$  parties  $p_1, \dots, p_n$  hold private inputs  $x_1, \dots, x_n$  respectively and engage in a protocol  $\Pi$  in order to compute (or approximate) a function  $f(\cdot)$  of their joint inputs. Parties are *honest-but-curious*, which means they follow the prescribed randomized protocol. However, as the execution of the protocol terminates, colluding parties can try to infer information about inputs of parties outside the coalition, given their joint view of the execution.

The protocol  $\Pi$  is executed in a synchronous environment with point-to-point secure (untappable) communication channels, and is required to preserve privacy with respect to coalitions of up to  $t$  parties. Following [5], we assume that the protocol  $\Pi$  has a *fixed-communication* pattern (such protocols are called *oblivious*), i.e., every channel is either (i) active in every run of  $\Pi$  (i.e., at least one bit is sent over the channel), or (ii) never used<sup>3</sup>. Parties that are adjacent to at least  $t + 1$  active channels are called *popular* other parties are called *lonely*.

The main definition we will work with is an extension of Definition 2.2 to a distributed setting. Informally, we require that differential privacy is preserved with respect to any coalition of size up to  $t$ .

<sup>3</sup>Our proofs also work in a relaxed setting where every channel is either (i) used in at least a constant fraction of the runs of  $\Pi$  (where the probability is taken over the coins of  $\Pi$ ), or (ii) is never used.

**Definition 2.4 (Distributed differential privacy)** Let  $\Pi$  be a protocol between  $n$  (honest-but-curious) parties. For a set  $T \subseteq [n]$  and fixed inputs  $\mathbf{x} = (x_1, \dots, x_n)$ , let  $\text{View}_T(x_1, \dots, x_n)$  be the random variable containing the inputs of the parties in  $T$  (i.e.,  $\{x_i\}_{i \in T}$ ), the random inputs of the parties in  $T$ , and the messages that the parties in  $T$  received during the execution of the protocol with private inputs  $\mathbf{x} = (x_1, \dots, x_n)$  (the randomness is taken over the random inputs of the parties not in  $T$ ).

We say that  $\Pi$  is  $(t, \varepsilon)$ -differentially private if for all  $T \subset [n]$ , where  $|T| \leq t$ , for all  $T$ -neighboring  $\mathbf{x}, \mathbf{x}'$ , and for all possible sets  $\mathcal{V}_T$  of views of the parties in  $T$ :

$$\Pr[\text{View}_T(\mathbf{x}) \in \mathcal{V}_T] \leq e^\varepsilon \cdot \Pr[\text{View}_T(\mathbf{x}') \in \mathcal{V}_T], \quad (2)$$

where the probabilities are taken over the random inputs of the parties in the protocol  $\Pi$ .

An equivalent requirement is that for all  $T \subset [n]$ , where  $|T| \leq t$ , for all  $T$ -neighboring  $\mathbf{x}, \mathbf{x}'$ , and for all distinguishers  $D$  (i.e., functions, not necessarily efficiently computable, from views to  $\{0, 1\}$ ),

$$\Pr[D(\text{View}_T(\mathbf{x})) = 1] \leq e^\varepsilon \cdot \Pr[D(\text{View}_T(\mathbf{x}')) = 1].$$

This requirement can be relaxed to only consider distinguishers that are computationally bounded:

**Definition 2.5 (Computational distributed differential privacy)** We say that  $\Pi$  is computationally  $(t, \varepsilon)$ -differentially private if for every probabilistic polynomial-time algorithm  $D$ , and for every polynomial  $p(\cdot)$ , there exists  $k_0$  such that for all  $k \geq k_0$ , for all  $T \subset [n]$ , where  $|T| \leq t$ , and for all  $T$ -neighboring inputs  $\mathbf{x}, \mathbf{x}' \in (\{0, 1\}^k)^n$ :

$$\Pr[D(\text{View}_T(\mathbf{x})) = 1] \leq e^\varepsilon \cdot \Pr[D(\text{View}_T(\mathbf{x}')) = 1] + \frac{1}{p(n \cdot k)},$$

where the probabilities are taken over the random inputs of the parties in protocol  $\Pi$  and the randomness of  $D$ .

**Example 2.6** We next describe a computationally  $(n/2, \varepsilon)$ -differentially private protocol for computing SUM with  $O(\log n/\varepsilon)$  additive error,  $O(n)$  messages, and constant number of rounds. This protocol uses a homomorphic encryption scheme with threshold decryption (that is, only the sets of all parties can decrypt messages). For example, if we use ElGamal encryption, the distributed key generation and decryption require one round in which each party sends one message. The protocol works in three phases:

**Key Generation.** The parties generate public and private keys for the homomorphic encryption scheme with threshold decryption.

**Encryption.** Each party  $p_i$  chooses a random noise  $e_i$  (according to a distribution that will be defined later), computes  $y_i = x_i + e_i$ , encrypts  $y_i$  using the public encryption key and sends the encryption to  $p_1$ .

**Decryption.** Party  $p_1$  computes  $z$ , an encryption of  $y = \sum_{i=1}^n y_i$  (this is possible as the encryption scheme is homomorphic).  $p_1$  sends  $z$  to each  $p_i$ , which in return sends a decryption message back to  $p_1$ . Finally,  $p_1$  decrypts  $y$  from the decryption messages and sends  $y$  to all parties.

One way to generate each party's noise is for each party to sample from the Normal distribution with mean zero and variance  $6 \log^2 n / (n\epsilon^2)$ . Since the sum of normal random variables is a normal random variable,  $y = \sum_{i=1}^n x_i + \text{noise}$  where noise is sampled from a normal distribution with mean zero and variance  $6 \log^2 n / \epsilon^2$ . Furthermore, even if a coalition of  $n/2$  parties subtracts the noise that its parties added to  $y$ , the variance of the remaining noise is  $3 \log^2 n / \epsilon^2$ . Using the analysis of [9], the protocol is a computationally  $(n/2, \epsilon)$ -differentially private protocol which with constant probability has error  $O(\log n / \epsilon)$ .

The above protocol is a computationally  $(n/2, \epsilon)$ -differentially private protocol for computing SUM with  $O(\log n / \epsilon)$  additive error,  $O(n)$  messages, and constant number of rounds. In contrast, we prove that  $(n/2, \epsilon)$ -differentially information-theoretically private protocol for computing SUM with  $o(\sqrt{n})$  additive error and constant number of rounds must send  $\Omega(n^2)$  messages. Thus, our results shows that requiring only computational differential-privacy does result in more efficient protocols.

Using standard SFE feasibility results (in the computational setting), it is possible now to prove that the natural paradigm presented in Section 1 yields protocols that adhere to Definition 2.5. Consider an  $\epsilon$ -differentially private data analysis  $\hat{f}$  and a computationally bounded distinguisher  $D$ , trying distinguish between a computation of an SFE protocol computing  $\hat{f}$  with neighboring inputs  $\mathbf{x}$  and  $\mathbf{x}'$ . Since,  $\hat{f}$  preserves differential privacy the distributions on the outputs must be  $\epsilon$  close, the same must hold for the random variables describing the adversary's view (up to some negligible function in the length of the (concatenated) inputs). We get:

**Lemma 2.7 (Informal)** *Let  $\hat{f}$  be  $\epsilon$ -differentially private, and let  $\Pi$  be a  $t$ -secure protocol computing  $\hat{f}$ , then  $\Pi$  is computationally  $(t, \epsilon)$ -differentially private.*

In the above lemma, the if the  $t$ -secure protocol  $\Pi$  computing  $\hat{f}$  has perfect security, then  $\Pi$  is information-theoretically  $(t, \epsilon)$ -differentially private.

**Remark 2.8** *We will only consider protocols computing a (randomized) function  $\hat{f}(\cdot)$  resulting in all parties computing the same outcome of  $\hat{f}(\mathbf{x})$ . This can be achieved, e.g., by having one party compute  $\hat{f}(\mathbf{x})$  and send the outcome to all other parties.*

### 2.3 Distributed Protocols – Basic Observations

The following notation and basic observations are used throughout the paper.

**Notation 2.9** *Fix an  $n$ -party randomized protocol  $\Pi$  and fix some communication transcript  $c$ . Assume that party  $p_i$  holds an input  $x_i$  and receives messages according to the transcript  $c$ . We define  $\alpha_i^c(x_i)$  to be the probability that on input  $x_i$  party  $p_i$  sends messages that are consistent with transcript  $c$ , given that it receives messages that are consistent with  $c$ . The probability is taken over the randomness of party  $p_i$ .*

Let  $\ell$  be the number of rounds in  $\Pi$ . Assume, without loss of generality, that  $p_i$  receives and sends messages in every round, and let  $\beta_j^c$  (where  $1 \leq j \leq \ell$ ) be the probability on input  $x_i$  party  $p_i$  sends in round  $j$  messages that are consistent with transcript  $c$ , provided that in previous rounds  $p_i$  sees messages that are consistent with  $c$ . Then, by the chain rule of conditional probabilities we have that

$$\alpha_i^c(x_i) = \prod_{j=1}^{\ell} \beta_j^c.$$



Observe that the event that  $p_i$  sends messages according to  $c$  when it sees messages according to  $c$  depends only on the randomness  $r_i$ , and hence this event is independent of whether the other parties send messages according to  $c$  when they see messages according to  $c$ . We hence get the following lemma:

**Lemma 2.10** *Fix an  $n$ -party randomized protocol  $\Pi$ , assume that each  $p_i$  holds an input  $x_i$ , and fix some communication transcript  $c$ . Then, the probability that  $c$  is exchanged is  $\prod_{i=1}^n \alpha_i^c(x_i)$ .*

## 2.4 The Local Model

The local model (previously discussed in [11, 19]) is a simplified distributed communication model where the parties communicate via a designated party – a *curator*<sup>4</sup> – denoted  $C$ . The curator has no local input. We will consider two types of differentially private local protocols – interactive and non-interactive.

In *non-interactive* local protocols each party  $p_i$  applies an  $\varepsilon$ -differentially private algorithm  $S_i$  on its private input  $x_i$  and randomness  $r_i$ , and sends  $S_i(x_i, r_i)$  to  $C$  that then performs an arbitrary computation and publishes its result.

In *interactive* local protocols the protocol proceeds in *rounds*, where in each round  $j$  the curator sends to each party  $p_i$  a “query” message  $q_{i,j}$  and party  $p_i$  responds with the  $j$ th “answer”  $A_i(x_i, q_{i,1}, \dots, q_{i,j}, r_i)$ ; the answer is a function of the party’s input  $x_i$ , its random input  $r_i$ , and the first  $j$  queries. I.e., each round consists of two communication phases: first, the query messages are sent by the curator, then, each party sends the appropriate response message.

We note that in the honest-but-curious setting we can assume, without loss of generality, that the curator is deterministic, as randomness for the curator may be provided by parties in their first message.

**Definition 2.11 (Differential privacy in the local model)** *We say that a protocol  $\Pi$  in the local model is  $\varepsilon$ -differentially private if the curator’s view preserves  $\varepsilon$ -differential privacy. Formally, for all neighboring  $\mathbf{x}, \mathbf{x}'$  and for every possible set  $\mathcal{V}_C$  of views of the curator:*

$$\Pr[\text{View}_C(\mathbf{x}) \in \mathcal{V}_C] \leq e^\varepsilon \cdot \Pr[\text{View}_C(\mathbf{x}') \in \mathcal{V}_C],$$

where  $\text{View}_C(\mathbf{x})$  is the random variable containing the messages that  $C$  receives during the execution of the protocol with private inputs  $\mathbf{x} = (x_1, \dots, x_n)$  and the probability is taken over the random inputs of the parties.

We note that  $\text{View}_C(\mathbf{x})$  is defined in accordance with Definition 2.4 (with some abuse of notation, as we write  $C$  instead of  $\{C\}$ ). However, since  $C$  has no initial input and since  $C$  is assumed to be deterministic, it is enough to include in  $\text{View}_C(\mathbf{x})$  only the messages that  $C$  receives during the execution of the protocol with inputs  $\mathbf{x} = (x_1, \dots, x_n)$ .

Differential privacy in the local model may be equivalently phrased as a requirement to preserve the privacy of each party independently of other parties. We next give a definition in this spirit by considering the probabilities that a party  $p_i$  replies in a certain way to a given sequence of queries with, say,  $x_i = 0$  and with, say,  $x_i = 1$ . Any communication transcript  $c$  in an execution of the protocol defines a transcript  $c_i$ , where

$$c_i = (q_{i,1}, a_{i,1}, \dots, q_{i,\ell}, a_{i,\ell})$$

is the restriction of  $c$  to the messages transferred between party  $p_i$  and the curator (recall that in the local model every party communicates solely with the curator). Thus, we can use  $\alpha_i^{c_i}(x_i)$  (see Notation 2.9) to

<sup>4</sup>Unlike in a centralized setting where the curator is a trusted party that collects raw private information, in the local model the curator is a non-trusted party. In our setting, the curator is semi-honest.

denote the probability that  $p_i$  with private input  $x_i$  replies by  $a_{i,1}, \dots, a_{i,\ell}$  provided the curator has sent queries  $q_{i,1}, \dots, q_{i,\ell}$ . Using this notation, we present the alternative definition of privacy in the local model.

**Definition 2.12 (Differential privacy in the local model by individual privacy)** *We say that a protocol  $\Pi$  in the local model is  $\varepsilon$ -differentially private if the curator's view preserves  $\varepsilon$ -differential privacy with respect to each party separately. Formally, for every  $i \in [n]$  and for any possible communication transcript  $c_i = (q_{i,1}, a_{i,1}, \dots, q_{i,\ell}, a_{i,\ell})$  between party  $p_i$  and the curator (i.e., there exist inputs  $x'_1, \dots, x'_n$  and random inputs  $r'_1, \dots, r'_n$  consistent with  $c_i$ ), and for every  $x_i, y_i \in D$  it holds that*

$$\alpha_i^{c_i}(x_i) \leq e^\varepsilon \cdot \alpha_i^{c_i}(y_i),$$

where the probabilities are taken over the random input of  $p_i$ .

**Claim 2.13** *Definition 2.11 is equivalent to Definition 2.12.*

**Proof:** We prove implications in both directions.

**Definition 2.11  $\Rightarrow$  Definition 2.12:** Let  $\Pi$  be according to Definition 2.11. Given a possible transcript  $c_i$  of messages between party  $p_i$  and  $C$ , choose any possible transcript  $c = (c_1, \dots, c_n)$  that is consistent with  $c_i$ . We get that for all  $x_i, y_i$ ,

$$\frac{\alpha_i^{c_i}(x_i)}{\alpha_i^{c_i}(y_i)} = \frac{\alpha_i^{c_i}(x_i)}{\alpha_i^{c_i}(y_i)} \cdot \frac{\prod_{j \neq i} \alpha_j^{c_j}(x_j)}{\prod_{j \neq i} \alpha_j^{c_j}(x_j)} = \frac{\Pr[\text{View}_C(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n) = c]}{\Pr[\text{View}_C(x_1, \dots, x_{i-1}, y_i, x_{i+1}, \dots, x_n) = c]} \leq e^\varepsilon,$$

where the last equality follows by Lemma 2.10, and the last inequality follows from  $\Pi$  being  $\varepsilon$ -differentially private according to Definition 2.11, noting that  $(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n)$  and  $(x_1, \dots, x_{i-1}, y_i, x_{i+1}, \dots, x_n)$  are neighboring.

**Definition 2.12  $\Rightarrow$  Definition 2.11:** Let  $\pi$  be according to Definition 2.12. Given a possible transcript  $c = (c_1, \dots, c_n)$  and neighboring inputs  $(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n)$  and  $(x_1, \dots, x_{i-1}, y_i, x_{i+1}, \dots, x_n)$ , we have that

$$\frac{\Pr[\text{View}_C(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n) = c]}{\Pr[\text{View}_C(x_1, \dots, x_{i-1}, y_i, x_{i+1}, \dots, x_n) = c]} = \frac{\alpha_i^c(x_i)}{\alpha_i^c(y_i)} \cdot \frac{\prod_{j \neq i} \alpha_j^c(x_j)}{\prod_{j \neq i} \alpha_j^c(x_j)} = \frac{\alpha_i^c(x_i)}{\alpha_i^c(y_i)} \leq e^\varepsilon,$$

where the first equality follows by Lemma 2.10 and the the inequality follows from  $\Pi$  being  $\varepsilon$ -differentially private according to Definition 2.12.  $\square$

Claim 2.13 implies that, in the information-theoretic local model, requiring differential privacy for the curator implies differential privacy with respect to every coalition.

## 2.5 Approximation

We will construct protocols whose outcome approximates a function  $f : D^n \rightarrow \mathbb{R}$  by a probabilistic function, according to the following definition:

**Definition 2.14 (Approximation)** *A randomized function  $\hat{f} : D^n \rightarrow \mathbb{R}$  is an additive  $(\gamma, \tau)$ -approximation for a (deterministic) function  $f$  if*

$$\Pr \left[ |f(\mathbf{x}) - \hat{f}(\mathbf{x})| > \tau(n) \right] \leq \gamma(n)$$

for all  $\mathbf{x} \in D^n$ . The probability is over the randomness of  $\hat{f}$ .

For example, by the properties of the Laplace distribution, Equation (1) yields an additive  $(e^{-k}, k \cdot \text{GS}_f/\varepsilon)$ -approximation to  $f$ , for every  $k > 0$ .

## 2.6 The Binary Sum and Gap Threshold Functions

The *binary sum* function is defined to be  $\text{SUM}_n(x_1, \dots, x_n) = \sum_{i=1}^n x_i$  for  $x_i \in \{0, 1\}$  (the subscript  $n$  is omitted when it is clear from the context). We will use a *gap* (or *promise*) version of the threshold function:

**Definition 2.15 (Gap Threshold)** For  $\kappa, \tau > 0$ ,

$$\text{GAP-TR}_{\kappa, \tau}(x_1, \dots, x_n) = \begin{cases} 0 & \text{If } \text{SUM}_n(x_1, \dots, x_n) \leq \kappa, \\ 1 & \text{If } \text{SUM}_n(x_1, \dots, x_n) \geq \kappa + \tau. \end{cases}$$

Note that  $\text{GAP-TR}_{\kappa, \tau}(x_1, \dots, x_n)$  is not defined when  $\kappa < \text{SUM}_n(x_1, \dots, x_n) < \kappa + \tau$ .

It is easy to transform any  $(\gamma, \tau/2)$ -approximation  $\hat{f}$  of  $\text{SUM}$  to a  $(\gamma, 0)$ -approximation  $\hat{g}$  to  $\text{GAP-TR}_{\kappa, \tau}$ : given  $y = \hat{f}(\mathbf{x})$  for  $\text{SUM}_n(\mathbf{x})$ , set the  $\hat{g}(\mathbf{x})$  to be 0 if  $y \leq \kappa + \tau/2$  and 1 otherwise. We get the following simple corollary:

**Corollary 2.16** *If there exists an  $\ell$ -round,  $(t, \varepsilon)$ -differentially private protocol (resp.  $\varepsilon$ -differentially private protocol in the local model) that  $(\gamma, \tau/2)$ -approximates  $\text{SUM}_n$  sending  $\rho$  messages, then for every  $\kappa$  there exists an  $\ell$ -round,  $(t, \varepsilon)$ -differentially private protocol (resp.  $\varepsilon$ -differentially private protocol in the local model) that correctly computes  $\text{GAP-TR}_{\kappa, \tau}$  with probability at least  $1 - \gamma$ , sending at most  $\rho$  messages.*

Specifically, non-existence of  $(t, \varepsilon)$ -differentially private protocols for computing  $\text{GAP-TR}_{0, \tau}$  correctly with  $n(t+1)/4$  messages implies that there exists no  $(t, \varepsilon)$ -differentially private protocols for computing  $\text{SUM}_n$  with  $n(t+1)/4$  messages and additive error magnitude  $\tau/2$ . The next claim asserts that the same non-existence also implies that, for any  $0 \leq \kappa \leq n - \tau$ , there exists no  $(t, \varepsilon)$ -differentially private protocol for computing  $\text{GAP-TR}_{\kappa, \tau}$  correctly with  $n(t+1)/8$  messages. Again, it applies to both the distributed and the local models.

**Claim 2.17** *If for some  $0 \leq \kappa \leq n - \tau$  there exists an  $\ell$ -round,  $(t, \varepsilon)$ -differentially private (respectively,  $\varepsilon$ -differentially private in the local model)  $n$ -party protocol that correctly computes  $\text{GAP-TR}_{\kappa, \tau}$  with probability at least  $\gamma$  sending at most  $\rho$  messages, then there exists an  $\ell$ -round,  $(t/2, \varepsilon)$ -differentially private (respectively,  $\varepsilon$ -differentially private in the local model)  $n/2$ -party protocol that correctly computes  $\text{GAP-TR}_{0, \tau}$  with probability at least  $\gamma$  sending at most  $\rho$  messages.*

**Proof:** For  $\kappa \leq n/2$ , given an  $n$ -party protocol  $\Pi$  that correctly computes  $\text{GAP-TR}_{\kappa, \tau}$ , define an  $n/2$ -party protocol  $\Pi'$  for computing  $\text{GAP-TR}_{0, \tau}$  by simulating parties  $p_{\frac{n}{2}+1}, \dots, p_n$  where  $x_{\frac{n}{2}+1}, \dots, x_{\frac{n}{2}+\kappa}$  are set to 1 and  $x_{\frac{n}{2}+\kappa+1}, \dots, x_n$  are set to 0. In the local model, a designated party, say  $p_1$ , can simulate these  $n/2$  parties. In the distributed model, we let each party  $p_i$  simulate party  $p_{i+n/2}$ .

Observe that in the distributed model any view  $v$  of a coalition  $T'$  of size  $t' \leq t/2$  in some execution of the resulting protocol, is exactly the view of the coalition  $T$  of size  $2t' \leq t$ , implied by  $T'$  (for  $p_i \in T'$  we have  $p_i, p_{i+n/2} \in T$ ), in the appropriate computation of the original protocol. Moreover, any  $T'$ -neighboring  $\mathbf{x}, \mathbf{x}'$  define  $T$ -neighboring  $\mathbf{xy}, \mathbf{x}'\mathbf{y}$  (where  $\mathbf{y} = 1^\kappa 0^{\frac{n}{2}-\kappa}$ ), such that  $\Pr[\text{View}_{T'}(\mathbf{xy}) = v] = \Pr[\text{View}_{T'}(\mathbf{x}) = v]$  and  $\Pr[\text{View}_T(\mathbf{x}'\mathbf{y}) = v] = \Pr[\text{View}_{T'}(\mathbf{x}') = v]$ . Thus, by the privacy of the original protocol, the resulting protocol is  $(t/2, \varepsilon)$ -differentially private.

For  $\kappa > n/2$ , we can use the construction above to compute  $\text{GAP-TR}_{n-\kappa-\tau, \tau}$ , by flipping all input bits (that is, changing 1 to 0 and vice-versa) before engaging in the execution, running the protocol, and finally flipping the result of the computation.  $\square$

### 3 Motivating Examples

We begin with two examples manifesting benefits of choosing an analysis together with a differentially private protocol for computing it. In the first example, this paradigm yields more efficient protocols than the natural paradigm; in the second example, it yields simpler protocols.

#### 3.1 Binary Sum – $\sqrt{n}$ Additive Error

We begin with a simple protocol for approximating  $\text{SUM}_n$  within  $O(\sqrt{n}/\varepsilon)$ -additive approximation. This protocol is well known as *Randomized Response* [25]. We describe the protocol in the (non-interactive) local model, and it can be easily translated to a two round (and  $2n$  messages)  $(n, \varepsilon)$ -differentially private distributed protocol by letting some arbitrarily designated party (say  $p_1$ ) play the role of  $C$ .

Let  $\text{flip}_\alpha(x)$  be a randomized bit flipping operator returning  $x$  with probability  $0.5 + \alpha$  and  $1 - x$  otherwise, where  $\alpha = \frac{\varepsilon}{4+2\varepsilon}$ . The protocol proceeds as follows:

1. Each party  $p_i$  with private input  $x_i \in \{0, 1\}$  sends  $z_i = \text{flip}_\alpha(x_i)$  to  $C$ .
2.  $C$  locally computes and publishes  $k = \sum_{i=1}^n z_i$ .
3. Each party locally computes  $\hat{f} = (k - (0.5 - \alpha)n)/2\alpha$ .

A total of  $O(n)$  messages and  $O(n \log n)$  bits of communication are exchanged. To see that the protocol satisfies the privacy requirement of Definition 2.12, note that

$$\frac{\Pr[\text{flip}_\alpha(1) = 1]}{\Pr[\text{flip}_\alpha(0) = 1]} = \frac{0.5 + \alpha}{0.5 - \alpha} = 1 + \varepsilon \leq e^\varepsilon,$$

and similarly  $\Pr[\text{flip}_\alpha(0) = 0]/\Pr[\text{flip}_\alpha(1) = 0] \leq e^\varepsilon$ . To see that the protocol approximates the sum function, note that

$$\mathbb{E}[z_i] = \mathbb{E}[\text{flip}_\alpha(x_i)] = \begin{cases} 0.5 + \alpha & \text{if } x_i = 1 \\ 0.5 - \alpha & \text{if } x_i = 0. \end{cases}$$

Thus,

$$\mathbb{E}[k] = (0.5 + \alpha) \cdot \text{SUM}(\mathbf{x}) + (0.5 - \alpha) \cdot (n - \text{SUM}(\mathbf{x})) = 2\alpha \cdot \text{SUM}(\mathbf{x}) + (0.5 - \alpha)n,$$

and hence,

$$\mathbb{E}[\hat{f}] = \mathbb{E}\left[\frac{k - (0.5 - \alpha)n}{2\alpha}\right] = \text{SUM}(\mathbf{x}).$$

By an application of the Chernoff bound, we get that  $\hat{f}$  is an additive  $(O(1), O(\sqrt{n}/\varepsilon))$ -approximation to  $\text{SUM}(\cdot)$ , that is, with constant probability, the error is  $O(\sqrt{n}/\varepsilon)$ .

**Remark 3.1** We next sketch an alternative  $\varepsilon$ -differentially private protocol that  $(O(1), \sqrt{n}/\varepsilon)$ -approximates  $\text{SUM}_n$ :

1. Each party  $p_i$  with private input  $x_i \in \{0, 1\}$  samples  $y_i \sim \text{Lap}(1/\varepsilon)$  and sends  $z_i = x_i + y_i$  to  $C$ .
2.  $C$  locally computes  $\hat{f} = \sum_{i=1}^n z_i$  and publishes the result.

The privacy of the protocol follows from the arguments in Section 2.1.

**Remark 3.2** *The above constructions result in symmetric approximations to  $\text{SUM}(\cdot)$  (i.e., the output distribution depends solely on  $\text{SUM}(\cdot)$  and not on the specific assignment). While these differentially private protocols use  $O(n)$  messages, it can be shown that for such symmetric functions that no efficient SFE protocols for such functions exist (see Section 6 for more details).*

### 3.2 Distance from a Long Subsequence of 0's

Our second function measures how many bits in a sequence  $\mathbf{x}$  of  $n$  bits should be set to zero to get an all-zero consecutive subsequence of length  $n^\alpha$ . In other words, the function should return the minimum weight over all substrings of  $\mathbf{x}$  of length  $n^\alpha$  bits:

$$\text{DIST}_\alpha(\mathbf{x}) = \min_i \left( \sum_{j=i}^{i+n^\alpha-1} x_j \right).$$

For  $t \leq n/2$  we present a  $(t, \varepsilon, \delta)$ -differentially private protocol<sup>5</sup> approximating  $\text{DIST}_\alpha(\mathbf{x})$  with additive error  $\tilde{O}(n^{\alpha/3}/\varepsilon)$ .

In our protocol, we treat the  $n$ -bit string  $\mathbf{x}$  (where  $x_i$  is held by party  $p_i$ ) as a sequence of  $n^{1-\alpha/3}$  disjoint intervals, each  $n^{\alpha/3}$  bit long. Let  $i_1, \dots, i_{n^{1-\alpha/3}}$  be the indices of the first bit in each interval, and observe that  $\min_{i_k} (\sum_{j=i_k}^{i_k+n^\alpha-1} x_j)$  is an  $n^{\alpha/3}$  additive approximation of  $\text{DIST}_\alpha$ . The protocol for computing an approximation  $\hat{f}$  to  $\text{DIST}_\alpha$  is sketched below.

1. Every party  $p_i$  generates a random variable  $Y_i$  distributed according to the normal distribution  $N(\mu = 0, \sigma^2 = 2R/n)$  where  $R = \frac{2 \log(\frac{2}{\delta})}{\varepsilon^2}$ , and shares  $x_i + Y_i$  between the parties  $p_1, \dots, p_{t+1}$  using an additive  $(t+1)$ -out-of- $(t+1)$  secret sharing scheme<sup>6</sup>.
2. Every party  $p_i$ , where  $1 \leq i \leq t+1$ , sums, for every interval of length  $n^{\alpha/3}$ , the shares it got from the parties in the interval and sends this sum to  $p_1$ .
3. For every interval of length  $n^{\alpha/3}$ , party  $p_1$  computes the sum of the  $t+1$  sums it got for the interval. By the additivity of the secret sharing scheme, this sum is equal to

$$S_k = \sum_{j=i_k}^{i_k+n^{\alpha/3}-1} (x_j + Y_j) = \sum_{j=i_k}^{i_k+n^{\alpha/3}-1} x_j + Z_k,$$

where  $Z_k = \sum_{j=i_k}^{i_k+n^{\alpha/3}-1} Y_j$  (notice that  $Z_k \sim N(\mu = 0, \sigma^2 = 2R)$ ).

4.  $p_1$  computes  $\min_k \sum_{j=k}^{k+n^{2\alpha/3}} S_k$  and sends this output to all parties.

Using the analysis of [9], this protocol is a  $(t, \varepsilon, \delta)$ -differentially private protocol when  $2t < n$ . Furthermore, it can be shown that with high probability the additive error is  $\tilde{O}(n^{\alpha/3}/\varepsilon)$ . To conclude, we showed a simple 3 round protocol for  $\text{DIST}_\alpha$ .

<sup>5</sup> $(\varepsilon, \delta)$ -differential privacy is a generalization, defined in [9], of  $\varepsilon$ -differential privacy where it is only required that  $\Pr[\hat{f}(\mathbf{x}) \in \mathcal{V}] \leq e^\varepsilon \cdot \Pr[\hat{f}(\mathbf{x}') \in \mathcal{V}] + \delta$ .

<sup>6</sup>Shared secrets are taken from a finite domain by rounding the numbers  $\log n$  digits after the point. This yields no breach in privacy and adds a small magnitude of error.

This protocol demonstrates two advantages of the paradigm of choosing what and how together. First, we choose an approximation of  $\text{DIST}_\alpha$  (i.e., we compute the minimum of subsequences starting at a beginning of an interval). This approximation reduces the communication in the protocol. Second, we leak information beyond the output of the protocol, as  $p_1$  learns the sums  $S_k$ 's<sup>7</sup>.

## 4 Lowerbounds on the Error of Binary Sum and Gap-Threshold in the Local Model

We prove that any  $\ell$ -round  $\varepsilon$ -differentially private protocol in the local model for computing the binary sum function must exhibit an additive error of  $\Omega(\sqrt{n}/\tilde{O}(\ell))$ . By Corollary 2.16 and Claim 2.17, it suffices to prove that such a protocol can only compute  $\text{GAP-TR}_{0,\tau}$  for  $\tau = \Omega(\sqrt{n}/\tilde{O}(\ell))$  (i.e., the parameter  $\kappa$  is set to zero). For that, we show that there are two input vectors – one containing  $\Omega(\sqrt{n})$  ones, and the other is all zero – for which the curator sees similar distributions on the messages, and hence must return similar answers.

We will begin by having the non-zero vector be distributed according to a probability distribution  $\mathcal{A}$  (on  $n$ -bit vectors). This implies that a specific choice for this vector exists. In the following we set

$$\alpha \triangleq \frac{1}{\varepsilon\sqrt{dn}}, \quad (3)$$

where  $d > 1$  (the value of  $d$ , which is a function of the number of rounds in the protocol  $\ell$ , is determined later).

**Notation 4.1** Define the distribution  $\mathcal{A}$  on inputs from  $\{0, 1\}^n$  as follows: a vector  $\mathbf{x} = (x_1, \dots, x_n)$  is chosen, where  $x_i = 1$  with probability  $\alpha$  and  $x_i = 0$  with probability  $(1 - \alpha)$  (each input  $x_i$  is chosen independently).

We use  $\mathbf{X}$  to identify the random variable representing the joint input and  $X_i$  for the random variable corresponding to its  $i$ -th coordinate. The notation  $\Pr_{\mathcal{A}}[\cdot]$  is used when a probability over the choice of  $\mathbf{X}$  from  $\mathcal{A}$  is considered. For a set  $D$  of possible curator's views we use the notation  $\Pr_{\mathcal{A}}[D]$  to denote the probability of the event that the view of the curator falls in  $D$  when the joint input  $\mathbf{X}$  is chosen according to  $\mathcal{A}$ .

**Main steps of the proof:** In Section 4.1, we analyze properties of non-interactive differentially private protocols in the local model, and show that a curator, trying to distinguish between an input chosen according to distribution  $\mathcal{A}$  and the all zero input, fails with constant probability. In Section 4.2 we generalize this analysis to interactive protocols in the local model. In Section 4.3 we complete the proof of the lowerbound on the gap-threshold function in the local model.

### 4.1 Differentially Private Protocols in the Non-Interactive Local Model

Consider protocols in the non-interactive local model where each party holds an input  $x_i \in \{0, 1\}$  and independently applies an algorithm  $S_i$  (also called a sanitizer) before sending the sanitized result  $c_i$  to the

---

<sup>7</sup>One can use the techniques of [6] to avoid leaking these sums while maintaining a constant number of rounds, however the resulting protocol is less efficient.

curator. We want to prove that if each  $S_i$  is  $2\varepsilon$ -differentially private for some  $0 < \varepsilon \leq 1$ <sup>8</sup>, then the curator errs with constant probability when trying to distinguish between an input chosen according to distribution  $\mathcal{A}$  and  $\mathbf{0}$  (where  $\mathbf{0}$  is the vector  $0^n$ )<sup>9</sup>.

For every possible view  $\mathbf{c} = (c_1, \dots, c_n)$  of the curator  $C$ , we consider the ratios of the probability of receiving messages according to  $c$  when the input is chosen according to  $\mathcal{A}$  and when it is  $\mathbf{0}$ . The probability is over the randomness of the protocol, and over the choice according to distribution  $\mathcal{A}$  where specified:

$$r(\mathbf{c}) \triangleq \frac{\Pr_{\mathcal{A}}[\text{View}_C(\mathbf{X}) = \mathbf{c}]}{\Pr[\text{View}_C(\mathbf{0}) = \mathbf{c}]} \quad \text{and} \quad r_i(c_i) \triangleq \frac{\Pr_{\mathcal{A}}[S_i(X_i) = c_i]}{\Pr[S_i(0) = c_i]}. \quad (4)$$

Since in a non-interactive protocol  $\Pr[\text{View}_C(\mathbf{0}) = \mathbf{c}] = \prod_{i=1}^n \Pr_{\mathcal{A}}[S_i(0) = c_i]$  (the sanitizers  $S_i$  use independent randomness) and  $\Pr_{\mathcal{A}}[\text{View}_C(\mathbf{X}) = \mathbf{c}] = \prod_{i=1}^n \Pr_{\mathcal{A}}[S_i(X_i) = c_i]$  (the sanitizers  $S_i$  use independent randomness and the entries of the random variable  $\mathbf{X}$  are chosen independently), we have that

$$r(\mathbf{c}) = \prod_{i=1}^n r_i(c_i). \quad (5)$$

We next show that if the inputs are selected according to  $\mathcal{A}$ , then with constant probability  $r(\mathbf{c})$  is bounded by a constant. In other words, for those views  $c$  of the curator that are likely when inputs are selected according to  $\mathcal{A}$ , the probability of seeing  $\mathbf{c}$  when the protocol is executed with inputs selected according to  $\mathcal{A}$  is similar to the probability of seeing  $\mathbf{c}$  when the protocol is executed with inputs set to zero.

Define a random variable  $\mathbf{C} = (C_1, \dots, C_n)$  where  $C_i = S_i(X_i)$  and  $X_i$  is chosen according to the distribution  $\mathcal{A}$ . Defining the random variables  $V_i \triangleq \ln r_i(C_i)$ , we can write for every  $\eta > 0$ :

$$\Pr_{\mathcal{A}}[r(\mathbf{C}) > \eta] = \Pr_{\mathcal{A}}\left[\prod_{i=1}^n r_i(C_i) > \eta\right] = \Pr_{\mathcal{A}}\left[\sum_{i=1}^n V_i > \ln \eta\right], \quad (6)$$

where the first equality is by (5) above. In the next two lemmas we show that each variable  $V_i$  is bounded, and bound its expectation. Both proofs use the  $2\varepsilon$ -differential privacy of the sanitizers. These bounds are then used with the Hoeffding bound in Lemma 4.5 where we bound  $\Pr_{\mathcal{A}}[r(\mathbf{C}) > \eta]$ .

**Lemma 4.2** *For every  $i$  and for any  $0 < \varepsilon \leq 1$ , with probability one,  $1 - 2\alpha\varepsilon \leq r(c_i) \leq 1 + 4\alpha\varepsilon$  and  $-4\alpha\varepsilon \leq V_i \leq 4\alpha\varepsilon$ .*

**Proof:** For every  $i$  and every value  $c_i$ ,

$$r_i(c_i) = \frac{\Pr_{\mathcal{A}}[S_i(X_i) = c_i]}{\Pr[S_i(0) = c_i]} = \frac{\alpha \Pr[S_i(1) = c_i] + (1 - \alpha) \Pr[S_i(0) = c_i]}{\Pr[S_i(0) = c_i]} = 1 + \alpha \left( \frac{\Pr[S_i(1) = c_i]}{\Pr[S_i(0) = c_i]} - 1 \right).$$

Using  $e^{-2\varepsilon} \leq \frac{\Pr[S_i(1)=c_i]}{\Pr[S_i(0)=c_i]} \leq e^{2\varepsilon}$ , we get that

$$1 + \alpha(e^{-2\varepsilon} - 1) \leq r_i(c_i) \leq 1 + \alpha(e^{2\varepsilon} - 1).$$

Using  $e^{2x} < 1 + 4x$  and  $1 - e^{-2x} < 2x$  for  $0 < x \leq 1$ , we get  $1 - 2\alpha\varepsilon \leq r_i(C_i) \leq 1 + 4\alpha\varepsilon$ . Recall that  $V_i = \ln r_i(C_i)$ . Using  $\ln(1 + x) \leq x$  and  $\ln(1 - x) \geq -2x$  for  $0 \leq x \leq 0.5$  and noting that  $\alpha = 1/(\varepsilon\sqrt{dn})$  and hence  $4\alpha\varepsilon \ll 0.5$ , we get that  $-4\alpha\varepsilon \leq V_i \leq 4\alpha\varepsilon$ .  $\square$

<sup>8</sup>We can relax the condition  $\varepsilon \leq 1$  by a condition  $\varepsilon \leq \varepsilon_0$  for any constant  $\varepsilon_0 \geq 1$ . This would affect some of the constants in the calculations below.

<sup>9</sup>We consider protocols that are  $2\varepsilon$ -differentially private to simplify the notation in Section 4.2.

**Lemma 4.3** For every  $i$  and for any  $0 < \varepsilon \leq 1$ ,

$$\mathbb{E}[V_i] \leq 32\alpha^2\varepsilon^2.$$

**Proof:** For the proof, we assume that the output of  $S_i$  is in a countable set. Let

$$B_b \triangleq \{c_i : r_i(c_i) = 1 + b\} \quad \text{for } -2\alpha\varepsilon \leq b \leq 4\alpha\varepsilon.$$

Lemma 4.2 implies that these are the only values possible for  $b$ . By the definition of  $r_i$ , for every  $c_i \in B_b$ ,

$$\frac{\Pr_{\mathcal{A}}[S_i(X_i) = c_i]}{\Pr[S_i(0) = c_i]} = r(c_i) = 1 + b,$$

and hence,

$$\Pr[S_i(0) \in B_b] = \frac{\Pr_{\mathcal{A}}[S_i(X_i) \in B_b]}{1 + b} \leq (1 - b + 2b^2) \cdot \Pr_{\mathcal{A}}[S_i(X_i) \in B_b]. \quad (7)$$

Let  $\beta = 2\alpha\varepsilon$ . We next bound  $\mathbb{E}[V_i]$ :

$$\begin{aligned} \mathbb{E}[V_i] &= \mathbb{E}_{\mathcal{A}}[\ln r(C_i)] = \sum_{-\beta \leq b \leq 2\beta} \Pr_{\mathcal{A}}[S_i(X_i) \in B_b] \cdot \ln(1 + b) \\ &\leq \sum_{-\beta \leq b \leq 2\beta} \Pr_{\mathcal{A}}[S_i(X_i) \in B_b] \cdot b \end{aligned} \quad (8)$$

$$= \sum_{-\beta \leq b \leq 2\beta} \Pr_{\mathcal{A}}[S_i(X_i) \in B_b] \cdot (1 + 2b^2) - \sum_{-\beta \leq b \leq 2\beta} \Pr_{\mathcal{A}}[S_i(X_i) \in B_b] \cdot (1 - b + 2b^2) \quad (9)$$

Where (8) follows by  $\ln(1+b) \leq b$ . Using (7) we can replace the second term in (9) by  $\sum_{-\beta \leq b \leq 2\beta} \Pr[S_i(0) \in B_b]$  and get

$$\begin{aligned} \mathbb{E}[V_i] &\leq (1 + 2(2\beta)^2) \sum_{-\beta \leq b \leq 2\beta} \Pr_{\mathcal{A}}[S_i(X_i) \in B_b] - \sum_{-\beta \leq b \leq 2\beta} \Pr[S_i(0) \in B_b] \\ &= (1 + 8\beta^2) \cdot \Pr_{\mathcal{A}}[S_i(X_i) \in \cup_b B_b] - \Pr[S_i(0) \in \cup_b B_b] \\ &\leq (1 + 8\beta^2) \cdot 1 - 1 = 8\beta^2 = 32\alpha^2\varepsilon^2. \end{aligned}$$

□

By Lemma 4.3,  $\mathbb{E}[\sum_{i=1}^n V_i] = \sum_{i=1}^n \mathbb{E}[V_i] \leq 32\alpha^2\varepsilon^2 n = 32/d$ . We next prove Lemma 4.5 which shows that  $\sum_{i=1}^n V_i$  is concentrated around this value. We use the Hoeffding bound:

**Theorem 4.4 (Hoeffding bound)** Let  $V_1, \dots, V_n$  be independent random variables such that  $V_i \in [a, b]$  and  $\sum_{i=1}^n \mathbb{E}[V_i] = \mu$ . Then, for every  $t > 0$ ,

$$\Pr \left[ \sum_{i=1}^n V_i - \mu \geq t \right] \leq \exp \left( -\frac{2t^2}{n(b-a)^2} \right).$$

**Lemma 4.5**  $\Pr_{\mathcal{A}}[r(\mathbf{C}) > \exp(\nu/d)] < \exp(-(\nu - 32)^2/32d)$  for every  $\nu > 32$ .



**Proof:** By Equation (6), Lemma 4.2, Lemma 4.3, and substituting  $\alpha = \frac{1}{\varepsilon\sqrt{dn}}$ :

$$\begin{aligned}
\Pr_{\mathcal{A}}[r(\mathbf{C}) > \exp(\nu/d)] &= \Pr_{\mathcal{A}}\left[\sum_{i=1}^n V_i > \frac{\nu}{d}\right] \\
&= \Pr_{\mathcal{A}}\left[\sum_{i=1}^n V_i - \sum_{i=1}^n \mathbb{E}V_i > \frac{\nu}{d} - \sum_{i=1}^n \mathbb{E}V_i\right] \\
&\leq \Pr_{\mathcal{A}}\left[\sum_{i=1}^n V_i - \sum_{i=1}^n \mathbb{E}V_i > \frac{\nu}{d} - n \cdot 32\alpha^2\varepsilon^2\right] \\
&\leq \exp\left(-\frac{2\left(\frac{\nu}{d} - n \cdot 32\alpha^2\varepsilon^2\right)^2}{64n\alpha^2\varepsilon^2}\right) \\
&= \exp\left(-(\nu - 32)^2/32d\right).
\end{aligned}$$

□

We now rephrase Lemma 4.5 in a way that would be more convenient for our argument in the next section. Let  $\Pi$  be a  $2\varepsilon$ -private, non-interactive, local protocol, where  $0 < \varepsilon \leq 1$ . For a possible curator's view  $\mathbf{c}$ , let

$$p_{\mathcal{A}}(\mathbf{c}) = \Pr_{\mathcal{A}}[\text{View}_C(\mathbf{X}) = \mathbf{c}] \quad \text{and} \quad p_0(\mathbf{c}) = \Pr[\text{View}_C(\mathbf{0}) = \mathbf{c}],$$

where in  $p_{\mathcal{A}}(\mathbf{c})$  the probability is taken over the choice of  $\mathbf{X}$  according to the distribution  $\mathcal{A}$  and the randomness of  $\Pi$ , and in  $p_0(\mathbf{c})$  the probability is taken over the randomness of  $\Pi$ . The following corollary follows from Lemma 4.5 and the definition of  $r$  in Equation (4).

**Corollary 4.6** *Assume we execute  $\Pi$  with input sampled according to distribution  $\mathcal{A}$ , then for every  $\nu > 32$ , with probability at least  $1 - \exp(-(\nu - 32)^2/32d)$ , the curator's view satisfies:*

$$p_{\mathcal{A}}(\mathbf{c}) \leq \exp(\nu/d) \cdot p_0(\mathbf{c}),$$

where the probability is taken over the random choice from  $\mathcal{A}$  and the randomness of  $\Pi$ .

## 4.2 Differentially Private Protocols in the Interactive Local Model

In this section we generalize Corollary 4.6 to interactive local protocols where each party holds an input  $x_i \in \{0, 1\}$ . The structure of our argument is as follows:

1. We decompose an  $\ell$ -round  $\varepsilon$ -differentially private protocol  $\Pi$  into  $\ell$  non-interactive, local protocols, and prove that each of the  $\ell$  protocols is  $2\varepsilon$ -differentially private. Thus, we can apply Corollary 4.6 to each protocol.
2. We view the original protocol as a protocol between the curator and a single party, simulating the other  $n$  parties. In this protocol the curator's goal is to determine whether inputs are all zero or they are sampled according to  $\mathcal{A}$ . We apply a composition lemma to show that the curator's success probability does not increase by too much as  $\ell$  grows. Clearly, this is true also for the original protocol.

### 4.2.1 A Composition Lemma

Consider an interactive protocol, where a (deterministic) curator  $C$  sends adaptive queries to a single (randomized) party  $p$  holding a private input  $x \in \{0, 1\}$  in a similar setup to that of the local model (except that we make no requirement for  $\varepsilon$ -differential privacy). We assume that the party  $p$  is stateless and that in each round  $1 \leq j \leq \ell$ , the protocol proceeds as follows:

1. In the first phase of round  $j$ , the curator  $C$  sends  $p$  a message  $q_j$  (this message is also called the query); this message is a function of the round number  $j$  and the messages the curator got from  $p$  in the previous rounds.
2. In the second phase of round  $j$ , party  $p$  chooses fresh random coins and based on these coins and the query  $q_j$  it computes a message  $\mathcal{V}_j$  and sends it to the curator. We consider the randomized function computing the message  $\mathcal{V}_j$  as an algorithm  $A_j$ , that is,  $\mathcal{V}_j = A_j(x)$ .

**Definition 4.7** We say that a possible outcome  $\mathcal{V}_j$  is  $\varepsilon$ -good for algorithm  $A_j$  if  $\Pr[A_j(1) = \mathcal{V}] \leq e^\varepsilon \Pr[A_j(0) = \mathcal{V}]$ , where the probabilities are taken over the randomness of algorithm  $A_j$ . An algorithm  $A_j$  is  $(\varepsilon, \delta)$ -good if  $\Pr[A_j(1) \text{ is } \varepsilon\text{-good for } A_j] \geq 1 - \delta$ , where the probability is taken over the randomness of  $A_j$ .

Let  $\Pi$  be a protocol, as defined above, in which for every  $j$  and every transcript of messages  $\mathcal{V}_1, \dots, \mathcal{V}_{j-1}$ , sent by  $p$  in rounds  $1, \dots, j-1$ , the curator  $C$  replies with a query  $q_j$ , such that the algorithm  $A_j$  resulting from  $q_j$  is an  $(\varepsilon, \delta)$ -good algorithm. Define a randomized algorithm  $\hat{A}$  that simulates the interaction between  $p$  and  $C$ , i.e., given input  $x \in \{0, 1\}$  it outputs a transcript  $(q_1, \mathcal{V}_1, q_2, \mathcal{V}_2, \dots, q_\ell, \mathcal{V}_\ell)$  sampled according to  $\Pi(x)$ .

**Lemma 4.8**  $\hat{A}$  is  $(\ell\varepsilon, 1 - (1 - \delta)^\ell)$ -good.

**Proof:** Choose a random transcript  $(q_1, \mathcal{V}_1, q_2, \mathcal{V}_2, \dots, q_\ell, \mathcal{V}_\ell)$ , and let  $A_1, A_2, \dots, A_\ell$  be the algorithms defined by this transcript. By our assumptions all these algorithms are  $(\varepsilon, \delta)$ -good. Thus, with probability at least  $(1 - \delta)^\ell$ , the transcript  $\hat{\mathcal{V}} = (q_1, \mathcal{V}_1, q_2, \mathcal{V}_2, \dots, q_\ell, \mathcal{V}_\ell)$  is such that  $\mathcal{V}_j$  is  $\varepsilon$ -good for  $A_j$  for all  $1 \leq j \leq \ell$ . It suffices, hence, to prove that when that happens the transcript  $\hat{\mathcal{V}}$  is  $\ell\varepsilon$ -good for  $\hat{A}$ , and indeed,

$$\begin{aligned}
\Pr[\hat{A}(1) = (q_1, \mathcal{V}_1, q_2, \mathcal{V}_2, \dots, q_\ell, \mathcal{V}_\ell)] &= \prod_{j=1}^{\ell} \Pr[A_j(1) = \mathcal{V}_j] \\
&\leq \prod_{j=1}^{\ell} e^\varepsilon \cdot \Pr[A_j(0) = \mathcal{V}_j] \\
&= e^{\ell\varepsilon} \cdot \prod_{j=1}^{\ell} \Pr[A_j(0) = \mathcal{V}_j] \\
&= e^{\ell\varepsilon} \cdot \Pr[\hat{A}(0) = (q_1, \mathcal{V}_1, q_2, \mathcal{V}_2, \dots, q_\ell, \mathcal{V}_\ell)].
\end{aligned}$$

The first and last equalities follow by independence and by the fact that the curator is deterministic. The inequality follows by the  $\ell$ -goodness of  $\mathcal{V}_1, \dots, \mathcal{V}_\ell$ .  $\square$

### 4.2.2 The Main Lemma

Let  $\Pi$  be an  $\ell$ -round, local,  $\varepsilon$ -differentially private protocol, where  $0 < \varepsilon \leq 1$ . For a possible curator's view  $\mathbf{c}$ , let

$$p_{\mathcal{A}}(\mathbf{c}) = \Pr_{\mathcal{A}}[\text{View}_C(\mathbf{X}) = \mathbf{c}] \quad \text{and} \quad p_0(\mathbf{c}) = \Pr[\text{View}_C(\mathbf{0}) = \mathbf{c}],$$

where in  $p_{\mathcal{A}}(\mathbf{c})$  the probability is taken over the choice of  $\mathbf{X}$  according to the distribution  $\mathcal{A}$  and the randomness of  $\Pi$ , and in  $p_0(\mathbf{c})$  the probability is taken over the randomness of  $\Pi$ .

**Lemma 4.9** *Assume we execute  $\Pi$  with input sampled according to distribution  $\mathcal{A}$ , then for every  $\nu > 32$ , with probability at least  $1 - \ell \cdot \exp(-(\nu - 32)^2/32d)$ , the curator's view satisfies:*

$$p_{\mathcal{A}}(\mathbf{c}) \leq \exp(\ell\nu/d) \cdot p_0(\mathbf{c}),$$

where the probability is taken over the random choice from distribution  $\mathcal{A}$  and the randomness of  $\Pi$ .

**Proof:** Recall that in the interactive local model, a protocol is composed of  $\ell$ -rounds where in each round the curator sends a query to each party and the party sends an answer. We modify the protocol, to make the parties stateless, by introducing the following changes to the interaction between the curator and every party  $p_i$ . Both changes do not affect the privacy of the protocol, nor its outcome.

1. In round  $j$  the curator sends all queries and answers  $q_1, a_1, \dots, a_{j-1}, q_j$  it sent and received from  $p_i$  in previous rounds<sup>10</sup>.
2. Party  $p_i$  chooses a fresh random string in each round, that is, in round  $j$ , party  $p_i$  chooses with uniform distribution a random string that is consistent with the queries and answers it got in the previous rounds (since we assume that the parties are computationally unbounded, such choice is possible). Party  $p_i$  uses this random string to answer the  $j$ th query. In other words, we can consider  $p_i$  as applying an algorithm  $A_j$  to compute the  $j$ th answer; this algorithm depends on the previous queries and answers and uses an independent random string  $r_j$ .

We next claim that  $A_j$  is  $2\varepsilon$ -differentially private. That is, we claim that the probability that  $a_j$  is generated given the previous queries and answers is roughly the same when  $p_i$  holds the bit 0 and when  $p_i$  holds the bit 1. For a transcript  $c$  of the first  $j$  rounds between  $p_i$  and the curator  $C$  and for  $x_i \in \{0, 1\}$ , we denote by  $R_c^{x_i}$  the set of all random strings  $r$ , such that  $p_i$  with private input  $x_i$  and random input  $r$  sends at each round messages according to  $c$ , provided it received all messages according to  $c$  in previous rounds. Recall that  $\Pr[r_j \in R_c^{x_i}]$  is denoted  $\alpha_i^c(x_i)$ . Let  $c_j = q_1, a_1, \dots, q_{j-1}, a_{j-1}, q_j, a_j$  be a  $j$ -round transcript and let  $c'_j = q_1, a_1, \dots, q_{j-1}, a_{j-1}, q_j$  be the prefix of  $c_j$  without the  $j$ th round answer  $a_j$  (that is,  $c_j = c'_j \circ a_j$ ). Note that, since  $r_j$  must be consistent with the  $c'_j$ , it holds for every  $x_i \in \{0, 1\}$  that  $\Pr[A_j(x_1) = a_j] = \Pr[r_j \in R_{c'_j}^{x_1} | r_j \in R_{c'_j}^{x_1}]$ . We therefore need to show that

$$e^{-2\varepsilon} \leq \frac{\Pr[A_j(1) = a_j]}{\Pr[A_j(0) = a_j]} = \frac{\Pr[r_j \in R_{c'_j}^1 | r_j \in R_{c'_j}^1]}{\Pr[r_j \in R_{c'_j}^0 | r_j \in R_{c'_j}^0]} \leq e^{2\varepsilon},$$

To show that, we use the following two facts, which follow from Definition 2.12:

<sup>10</sup>To simplify notation, we omit the subscript  $i$  from the queries and answers.

$$e^{-\varepsilon} \leq \frac{\alpha_i^{c_j}(1)}{\alpha_i^{c_j}(0)} = \frac{\Pr[r_j \in R_{c_j}^1]}{\Pr[r_j \in R_{c_j}^0]} \leq e^\varepsilon, \quad (10)$$

and

$$e^{-\varepsilon} \leq \frac{\alpha_i^{c'_j}(1)}{\alpha_i^{c'_j}(0)} = \frac{\Pr[r_j \in R_{c'_j}^1]}{\Pr[r_j \in R_{c'_j}^0]} \leq e^\varepsilon \quad (11)$$

Hence, we have

$$\begin{aligned} r &\triangleq \frac{\Pr[A_j(1) = a_j]}{\Pr[A_j(0) = a_j]} = \frac{\Pr[r_j \in R_{c_j}^1 \wedge r_j \in R_{c'_j}^1]}{\Pr[r_j \in R_{c_j}^1]} \cdot \frac{\Pr[r_j \in R_{c'_j}^0]}{\Pr[r_j \in R_{c_j}^0 \wedge r_j \in R_{c'_j}^0]} \\ &= \frac{\Pr[r_j \in R_{c_j}^1]}{\Pr[r_j \in R_{c'_j}^1]} \cdot \frac{\Pr[r_j \in R_{c'_j}^0]}{\Pr[r_j \in R_{c_j}^0]} = \frac{\alpha_i^{c_j}(1)}{\alpha_i^{c_j}(0)} \cdot \frac{\alpha_i^{c'_j}(0)}{\alpha_i^{c'_j}(1)}. \end{aligned}$$

Now, by using the right inequality in Equation (10) and the left inequality in Equation (11), we get that  $r \leq e^{2\varepsilon}$  and similarly, by using the left inequality in Equation (10) and the right inequality in Equation (11), we get that  $r \geq e^{-2\varepsilon}$ . Thus, the answers of the  $n$  parties in round  $j$  are  $2\varepsilon$ -private, and we can apply Corollary 4.6 to the concatenation of the  $n$  answers.

We now use the above protocol to construct a protocol  $\Pi_1$  between a single party, holding a one bit input  $x$  and a curator. Throughout the execution of the protocol the party simulates all  $n$  parties as specified by the original protocol  $\Pi$  (i.e., sends messages to the curator with the same distribution as the  $n$  parties send them). If the bit of the party in  $\Pi_1$  is 1 it chooses the  $n$  input bits of the  $n$  parties in  $\Pi$  according to distribution  $\mathcal{A}$ . If the bit of the party in  $\Pi_1$  is 0 it chooses the  $n$  input bits of the  $n$  parties in  $\Pi$  to be the all-zero vector. By Corollary 4.6 we can apply the composition lemma – Lemma 4.8 – to the composition of the  $\ell$ ,  $2\varepsilon$ -differentially private, non-interactive protocols and the lemma follows.  $\square$

**Corollary 4.10** *Let  $0 < \varepsilon \leq 1$ . For every  $\nu > 32$  and for every set  $D$  of views in an  $\ell$ -round,  $\varepsilon$ -differentially private, local protocol,*

$$\Pr[\text{View}_C(\mathbf{0}) \in D] \geq \frac{\Pr_{\mathcal{A}}[\text{View}_C(\mathbf{X}) \in D] - \ell \cdot \exp(-(\nu - 32)^2/32d)}{\exp(\ell\nu/d)}.$$

**Proof:** Let

$$D_1 = \left\{ \mathbf{c} \in D : \Pr_{\mathcal{A}}[\text{View}_C(\mathbf{X}) = \mathbf{c}] \leq \exp(\ell\nu/d) \Pr[\text{View}_C(\mathbf{0}) = \mathbf{c}] \right\}$$

and

$$D_2 = \left\{ \mathbf{c} \in D : \Pr_{\mathcal{A}}[\text{View}_C(\mathbf{X}) = \mathbf{c}] > \exp(\ell\nu/d) \Pr[\text{View}_C(\mathbf{0}) = \mathbf{c}] \right\}.$$

That is,  $D_2 = D \setminus D_1$ . By Lemma 4.9,  $\Pr_{\mathcal{A}}[\text{View}_C(\mathbf{X}) \in D_2] \leq \ell \exp(-(\nu - 32)^2/32d)$ , and, furthermore,  $\Pr[\text{View}_C(\mathbf{0}) \in D_1] \geq \Pr_{\mathcal{A}}[\text{View}_C(\mathbf{X}) \in D_1] / \exp(\ell\nu/d)$ . Thus,

$$\begin{aligned} \Pr[\text{View}_C(\mathbf{0}) \in D] &\geq \Pr[\text{View}_C(\mathbf{0}) \in D_1] \geq \frac{\Pr_{\mathcal{A}}[\text{View}_C(\mathbf{X}) \in D_1]}{e^{\ell\nu/d}} \\ &= \frac{\Pr_{\mathcal{A}}[\text{View}_C(\mathbf{X}) \in D] - \Pr_{\mathcal{A}}[\text{View}_C(\mathbf{X}) \in D_2]}{e^{\ell\nu/d}} \\ &\geq \frac{\Pr_{\mathcal{A}}[\text{View}_C(\mathbf{X}) \in D] - \ell e^{-(\nu-32)^2/32d}}{e^{\ell\nu/d}}. \end{aligned}$$

$\square$

### 4.3 Completing the Lowerbound for Gap-Threshold and Sum in the Local Model

We now complete the proof that in any  $\ell$ -round,  $\varepsilon$ -differentially private, local protocols for the gap-threshold function, namely,  $\text{GAP-TR}_{0,\tau}$ , if  $\tau \ll \sqrt{n}$  and  $\ell$  is small, then the curator errs with constant probability.

Recall that we constructed the distribution  $\mathcal{A}$  in which each bit in the input is chosen (independently at random) to be one with probability  $\alpha$  and zero with probability  $1 - \alpha$ . Lemma 4.11, which follows from a standard Chernoff bound argument, states that when generating a vector  $(X_1, \dots, X_n)$  according to  $\mathcal{A}$ , the sum  $\sum_{i=1}^n X_i$  is concentrated around its expected value, which is  $\alpha n$  (recall that  $\alpha = 1/(\varepsilon\sqrt{dn})$ ). We apply the following Chernoff bound: Given  $n$  zero-one random variables  $X_1, \dots, X_n$  and  $0 < t < 1$ ,  $\Pr[\sum_{i=1}^n X_i \leq (1-t)\mu] < \exp\left(-\frac{t^2\mu}{2}\right)$ , where  $\mu = \sum_{i=1}^n \mathbb{E}[X_i]$ .

**Lemma 4.11**  $\Pr_{\mathcal{A}}[\sum_{i=1}^n X_i \leq (1-\gamma)\alpha n] < \exp\left(-\frac{\gamma^2\sqrt{n}}{2\varepsilon\sqrt{d}}\right)$  for every  $0 < \gamma < 1$ .

**Proof:** We use the above bound with  $\mu = \alpha n = \frac{\sqrt{n}}{\varepsilon\sqrt{d}}$ . Thus,

$$\begin{aligned} \Pr_{\mathcal{A}}\left[\sum_{i=1}^n X_i \leq (1-\gamma)\alpha n\right] &< \exp\left(-\frac{\alpha n \gamma^2}{2}\right) \\ &< \exp\left(-\frac{\gamma^2\sqrt{n}}{2\varepsilon\sqrt{d}}\right). \end{aligned}$$

□

On one hand, by Corollary 4.10, the distributions on the outputs when the input vector is taken from  $\mathcal{A}$  and when it is the all zero vector are *not* far apart. On the other hand, by Lemma 4.11, with high probability the number of ones in the inputs distributed according to  $\mathcal{A}$  is fairly big. These facts are used in Theorem 4.12 to prove the lowerbound.

**Theorem 4.12** *Let  $0 < \varepsilon \leq 1$ . There exist constants  $c > 0$  and  $p > 0$  such that in any  $\ell$ -round,  $\varepsilon$ -differentially private, local protocol for computing  $\text{GAP-TR}_{0,\tau}$  for  $\tau = c\frac{\sqrt{n}}{\varepsilon\ell\sqrt{\log \ell}}$  the curator errs with probability at least  $p$ .*

**Proof:** Fix any  $\ell$ -round,  $\varepsilon$ -differentially private, local protocol for computing  $\text{GAP-TR}_{0,\tau}$ . Recall that in the local model the curator is assumed to be deterministic. Thus, the curator, having seen its overall view of the execution of the protocol  $c$ , applies a deterministic algorithm  $G$  to  $c$ , where  $G(c)$  is the output of the protocol (which supposed to answer  $\text{GAP-TR}_{0,\tau}(x_1, \dots, x_n)$  correctly). Let  $\tau = \alpha n/2 = \sqrt{n}/(2\varepsilon\sqrt{d})$ .

Denote by  $D$  the set vectors of communication for which the curator answers 1, i.e.,  $D \triangleq \{c : G(c) = 1\}$ . The idea of the proof is as follows. If the probability of  $D$  under the distribution  $\mathcal{A}$  is small, then the curator has a big error when the inputs are distributed according to  $\mathcal{A}$ . Otherwise, by Corollary 4.10, the probability of  $D$  when the inputs are all zero is big, hence the curator has a big error when the inputs are the all-zero string. Formally, there are two cases:

**Case I:**  $\Pr_{\mathcal{A}}[D] < 0.99$ . We consider the event that the sum of the inputs is at least  $\tau = \alpha n/2$  and the curator returns zero as an answer, that is, the curator errs.

We show that when the inputs are distributed according to  $\mathcal{A}$  the probability of the complementary of this event is bounded away from 1. By the union bound the probability of the complementary event is

at most  $\Pr_{\mathcal{A}}[\sum_{i=1}^n X_i < 0.5\alpha n] + \Pr_{\mathcal{A}}[D]$ . By Lemma 4.11,

$$\Pr_{\mathcal{A}}[D] + \Pr_{\mathcal{A}}\left[\sum_{i=1}^n X_i < 0.5\alpha n\right] \leq 0.99 + \exp\left(-0.25\sqrt{n}/(2\varepsilon\sqrt{d})\right) \approx 0.99.$$

Thus, in this case, with probability  $\approx 0.01$  the curator errs.

**Case II:**  $\Pr_{\mathcal{A}}[D] \geq 0.99$ . Here, we consider the event that the input is the all-zero string and the curator answers 1, that is, the curator errs.

We use Corollary 4.10 and show that when the inputs are all zero, the probability of this event is bounded away from 0 when taking  $\nu = \theta(\ell \log \ell)$  and  $d = \ell\nu = \theta(\ell^2 \log \ell)$ ,

$$\Pr[\text{View}_C(\mathbf{0}) \in D] \geq \frac{\Pr_{\mathcal{A}}[D] - \ell \exp\left(-(\nu - 32)^2/32d\right)}{\exp(\ell\nu/d)} > \frac{0.99 - 0.5}{\exp(1)} > 0.01.$$

Thus, in this case, with probability at least 0.01, the curator errs. As  $d = \theta(\ell^2 \log \ell)$ , we get that  $\tau = \sqrt{n}/(2\varepsilon\sqrt{d}) = \theta(\sqrt{n}/(\varepsilon\ell\sqrt{\log \ell}))$ .

□

By applying the local model variant of Corollary 2.16, we get our lowerbound for  $\text{SUM}_n$  as a corollary of Theorem 4.12:

**Corollary 4.13** *Let  $0 < \varepsilon \leq 1$ . There exist constants  $\delta > 0$  and  $p > 0$  such that in any  $\ell$ -round,  $\varepsilon$ -differentially private, local protocol for computing  $\text{SUM}_n$  the curator errs with probability at least  $p$  by at least  $\frac{\delta\sqrt{n}}{\varepsilon\ell\sqrt{\log \ell}}$ .*

**Proof:** Let  $\Pi$  be an  $\ell$ -round,  $\varepsilon$ -differentially private, local protocol for computing  $\text{SUM}_n$ , for which the curator errs by at most  $\tau$  with probability at most  $p$ . By Corollary 2.16 there exists an  $\ell$ -round,  $\varepsilon$ -differentially private, local protocol for computing  $\text{GAP-TR}_{0,2\tau}$  errs with probability at most  $p$ . □

## 5 Lowerbounds for Binary Sum and Gap-Threshold in the Distributed Model

We prove that, in any  $\ell$ -round, fixed-communication,  $(t, \varepsilon)$ -differentially private protocol computing the binary sum with additive error less than  $\sqrt{n}/\tilde{O}(\ell)$ , the number of messages sent in the protocol is  $\Omega(nt)$ . In the heart of our proof is the more general observation that in the information theoretic setting, a party that has at most  $t$  neighbors must protect its privacy with respect to his neighbors, since this set separates it from the rest of the parties. Thus, any such party, is essentially as limited as any party participating in a protocol in the local communication model.

### 5.1 From Distributed to Local Protocols

We start with the transformation of a distributed protocol, using a small number of messages to a protocol in the local model.

**Lemma 5.1** *If there exists an  $\ell$ -round, fixed communication,  $(t, \varepsilon)$ -differentially private protocol that  $(\gamma, \tau)$ -approximates  $\text{SUM}_n$  sending at most  $n(t+1)/4$  messages, then there exists an  $(\ell+1)$ -round,  $\varepsilon$ -differentially private protocol in the local model that  $(\gamma, \tau)$ -approximates  $\text{SUM}_{n/2}$ .*

**Proof:** The intuition behind the proof is that in the information theoretic model if an input of a party affects the output, then the neighbors of this party must learn information on its input. Recall that a party in a protocol  $\Pi$  is lonely if it communicates with at most  $t$  other parties and it is popular otherwise. If a party  $p_i$  is lonely then it has most  $t$  neighbors, thus, from the privacy requirement in  $(t, \varepsilon)$ -differentially private protocols, they are not allowed to learn “too much” information on the input of  $p_i$ . Therefore, the inputs of lonely parties cannot affect the output of the protocol by too much, thus, since there are many lonely parties, the protocol must have a large error.

Formally, assume that there is a distributed protocol  $\Pi$  satisfying the conditions in the lemma. As the protocol sends at most  $n(t + 1)/4$  messages, the protocol uses at most  $n(t + 1)/4$  channels. Since each channel connects two parties, there are at least  $n/2$  lonely parties. We will construct a protocol in the local model which  $(\gamma, \tau)$ -approximates  $\text{SUM}_{n/2}$  in two steps: In the first step, which is the main part of the proof, we construct a protocol  $\mathcal{P}$  in the local model which  $(\gamma, \tau)$ -approximates  $\text{SUM}_n$  and only protects the privacy of the lonely parties. In the second step, we fix the inputs of the popular parties and obtain a protocol  $\mathcal{P}'$  for  $n/2$  parties that protects the privacy of all parties.

**First Step.** We convert the distributed protocol  $\Pi$  to a protocol  $\mathcal{P}$  in the local model as follows: Recall that in the local model each round consists of two phases where in the first phase the curator sends queries to the parties and in the second phase parties send the appropriate responses. We hence have a single round in  $\mathcal{P}$  for every round of  $\Pi$  such that every message  $m$  that Party  $p_j$  sends to Party  $p_k$  in round  $i$  in protocol  $\Pi$ , Party  $p_j$  sends  $m$  to the curator in round  $i$  and the curator sends  $m$  to Party  $p_k$  in the first phase of round  $i + 1$ . Finally, at the end of the protocol Party  $p_1$  sends the output to the curator.

We next prove that  $\mathcal{P}$  protects the privacy of lonely parties. Without loss of generality, let  $p_1$  be a lonely party, let  $T$  be the set of size at most  $t$  containing the neighbors of  $p_1$ , and let  $R = \{p_1, \dots, p_n\} \setminus (T \cup \{p_1\})$ . See Figure 1 for a description of these sets. Fix any neighboring vectors of inputs  $\mathbf{x}$  and  $\mathbf{x}'$  which differ on  $x_1$ . The view of the curator in  $\mathcal{P}$  contains all messages sent in the protocol. It suffices to prove that for every view  $v$ ,

$$\Pr[\text{View}_C^{\mathcal{P}}(\mathbf{x}) = v] \leq e^\varepsilon \cdot \Pr[\text{View}_C^{\mathcal{P}}(\mathbf{x}') = v] \quad (12)$$

(by simple summation it will follow for every set of views  $\mathcal{V}$ ).

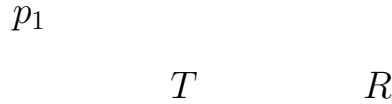


Figure 1: The partition of the parties to sets.

Fix a view  $v$  of the curator. For a set  $A$ , define  $\alpha_A$  and  $\alpha'_A$  as the probabilities in  $\Pi$  that in each round the set  $A$  with inputs from  $\mathbf{x}$  and  $\mathbf{x}'$  respectively sends messages according to  $v$  if it gets messages according to  $v$  in previous rounds (these probabilities are taken over the random inputs of the parties in  $A$ ). Observe that if  $p_1 \notin A$ , then  $\alpha_A = \alpha'_A$  (since  $\mathbf{x}$  and  $\mathbf{x}'$  only differ on  $x_1$ ). By simulating  $p_1, T, R$  by three parties

and applying Lemma 2.10, and by the construction of  $\mathcal{P}$  from  $\Pi$

$$\begin{aligned}\Pr[\text{View}_{\mathcal{C}}^{\mathcal{P}}(\mathbf{x}) = v] &= \alpha_{\{p_1\}} \cdot \alpha_T \cdot \alpha_R, \quad \text{and} \\ \Pr[\text{View}_{\mathcal{C}}^{\mathcal{P}}(\mathbf{x}') = v] &= \alpha'_{\{p_1\}} \cdot \alpha'_T \cdot \alpha'_R = \alpha'_{\{p_1\}} \cdot \alpha_T \cdot \alpha_R.\end{aligned}$$

Thus, proving Equation (12) is equivalent to proving that

$$\alpha_{\{p_1\}} \leq e^\varepsilon \alpha'_{\{p_1\}}. \quad (13)$$

We use the  $(t, \varepsilon)$ -privacy of protocol  $\Pi$  to prove Equation (13). Let  $v_T$  be the messages sent and received by the parties in  $T$  in  $v$ . As  $T$  separates  $p_1$  from  $R$ , the messages in  $v_T$  are all messages in  $v$  except for the messages exchanged between parties in  $R$ . The view of  $T$  includes the inputs of  $T$  in  $\mathbf{x}$ , the messages  $v_T$ , and the random inputs  $\mathbf{r}_T = \{r_i : p_i \in T\}$ . For a set  $A$ , define  $\beta_A$  and  $\beta'_A$  as the probabilities that in  $\Pi$  in each round the set  $A$  with inputs from  $\mathbf{x}$  and  $\mathbf{x}'$  respectively sends messages according to  $v_T$  if it gets messages according to  $v_T$  in previous rounds. Note that  $\beta_{\{p_1\}} = \alpha_{\{p_1\}}$  and  $\beta'_{\{p_1\}} = \alpha'_{\{p_1\}}$  by the definition of  $\mathcal{P}$ . By simulating  $p_1, T, R$  by three parties, where the random inputs of  $T$  are fixed to  $\mathbf{r}_T$ , and by Lemma 2.10,

$$\begin{aligned}\Pr[\text{View}_T^{\Pi}(\mathbf{x}) = (\mathbf{x}_T, \mathbf{r}_T, v_T)] &= \alpha_{\{p_1\}} \cdot \beta_R, \quad \text{and} \\ \Pr[\text{View}_T^{\Pi}(\mathbf{x}') = (\mathbf{x}_T, \mathbf{r}_T, v_T)] &= \beta'_{\{p_1\}} \cdot \beta'_R = \alpha'_{\{p_1\}} \cdot \beta_R.\end{aligned}$$

(recalling that  $\mathbf{x}_T = \mathbf{x}'_T$ ). The above probabilities are taken over the random strings of  $R$  and  $p_1$  when the random strings of  $T$  are fixed to  $\mathbf{r}_T$ . The  $(t, \varepsilon)$ -differential privacy of  $\Pi$  implies that

$$\Pr[\text{View}_T^{\Pi}(\mathbf{x}) = (\mathbf{x}_T, \mathbf{r}_T, v_T)] \leq e^\varepsilon \Pr[\text{View}_T^{\Pi}(\mathbf{x}') = (\mathbf{x}_T, \mathbf{r}_T, v_T)].$$

Thus,  $\alpha_{\{p_1\}} \leq e^\varepsilon \alpha'_{\{p_1\}}$  and, therefore,  $\mathcal{P}$  is  $\varepsilon$ -differentially private with respect to inputs of lonely parties.

**Second Step.** There are at least  $n/2$  lonely parties in  $\Pi$ ; without loss of generality, parties  $p_1, \dots, p_{n/2}$  are lonely. We construct a protocol  $\mathcal{P}'$  that  $(\gamma, \tau)$ -approximates  $\text{SUM}_{n/2}$  by executing Protocol  $\mathcal{P}$  where (i) Party  $p_i$ , where  $1 \leq i \leq n/2$ , with input  $x_i$  sends messages in  $\mathcal{P}'$  as the party  $p_i$  with input  $x_i$  sends them in  $\mathcal{P}$ ; and (ii) In addition, the party  $p_1$  in  $\mathcal{P}'$  simulates all other  $n/2$  parties in  $\mathcal{P}$ , that is, for every  $n/2 < i \leq n$ , it chooses a random input  $r_i$  for  $p_i$  and in every round it sends to the curator the same messages as  $p_i$  would send with  $x_i = 0$  and  $r_i$ . Since the curator sees the same view in  $\mathcal{P}$  and  $\mathcal{P}'$  and since the privacy of lonely parties is protected in  $\mathcal{P}$ , the privacy of each of the  $n/2$  parties in  $\mathcal{P}'$  is protected. Protocol  $\mathcal{P}'$ , therefore,  $(\gamma, \tau)$ -approximates  $\text{SUM}_{n/2}$  (since we fixed  $x_i = 0$  for  $n/2 < i \leq n$  and  $\mathcal{P}'$  returns the same output distribution of  $\Pi$ , which  $(\gamma, \tau)$ -approximates  $\text{SUM}_n$  for all possible inputs).  $\square$

We are now ready to state the main theorem of this section.

**Theorem 5.2** *Let  $0 < \varepsilon \leq 1$ . There exist constants  $\delta > 0$  and  $\gamma > 0$  such that in any  $\ell$ -round, fixed-communication,  $(t, \varepsilon)$ -differentially private protocol for approximating  $\text{SUM}_n$  that sends at most  $n(t+1)/4$  messages the protocol errs with probability at least  $\gamma$  by at least  $\frac{\delta\sqrt{n}}{\varepsilon\ell\sqrt{\log \ell}}$ .*

**Proof:** Assume, for sake of contradiction, that there is an  $\ell$ -round,  $(t, \varepsilon)$ -differentially private protocol  $\Pi$  for computing  $\text{SUM}_n$ , which sends at most  $n(t+1)/4$  messages and errs by at most  $\tau = \frac{\delta\sqrt{n}}{\varepsilon\ell\sqrt{\log \ell}}$  with probability at least  $1 - \gamma$ . By Lemma 5.1 there exists an  $(\ell+1)$ -round,  $\varepsilon$ -differentially private, local protocol



$\mathcal{P}$  for computing  $\text{SUM}_{n/2}$  which errs by at most  $\tau = \frac{\delta\sqrt{n}}{\varepsilon\ell\sqrt{\log\ell}} = \frac{\sqrt{2}\delta\sqrt{n/2}}{\varepsilon\ell\sqrt{\log\ell}}$  with probability at least  $1 - \gamma$ . This contradicts Corollary 4.13.  $\square$

Theorem 5.2 suggests that whenever we require that the error of a differentially private protocol for approximating SUM to be of magnitude smaller than  $\sqrt{n}/\varepsilon$ , there is no reason to relinquish the simplicity of the natural paradigm for constructing protocols. In this case, it is possible to construct relatively simple efficient SFE protocols, which use  $O(nt)$  messages, and compute an additive  $(O(1/\varepsilon), O(1))$ -approximation of SUM.

**Remark 5.3** *It can also be shown that in any  $\ell$ -round, fixed-communication,  $(t, \varepsilon)$ -differentially private protocol computing the  $\text{GAP-TR}_{\kappa, \tau}$ , for any  $0 \leq \kappa \leq n - \tau$ , the number of messages sent in the protocol is  $\Omega(nt)$ , for  $\tau = \sqrt{n}/\tilde{O}(\ell)$ . To show this, use the ideas similar to those of Lemma 5.1 and apply Theorem 4.12 to assert that any  $\ell$ -round, fixed-communication,  $(t, \varepsilon)$ -differentially private protocol computing the  $\text{GAP-TR}_{0, \tau}$ , the number of messages sent in the protocol is  $\Omega(nt)$ , for  $\tau = \sqrt{n}/\tilde{O}(\ell)$ . Then, using Claim 2.17, infer that the same is true for general  $\kappa$ .*

## 6 SFE for Symmetric Approximations of Binary-Sum

In this section we show the advantage of using the alternative paradigm for constructing distributed differentially private protocols whenever we allow an  $O(\sqrt{n}/\varepsilon)$  approximation. Recall that it is possible to construct differentially private protocols for such approximations that use  $2n$  messages and are secure against coalitions of size up to  $t = n - 1$  (see Section 3.1). We next prove, using ideas from Chor and Kushilevitz [5], that any SFE protocol for computing a symmetric approximation for  $\text{SUM}_n$ , using less than  $nt/4$  messages, has error magnitude  $\Omega(n)$ .

We first give the definition of SFE protocols computing a given randomized function  $\hat{f}(\cdot)$ . Here, again, we only consider protocols where all parties are honest-but-curious and compute the *same* output. The definition is given in the information-theoretic model.

**Definition 6.1 (SFE)** *Let  $\hat{f} : (\{0, 1\}^*)^n \rightarrow \{0, 1\}^*$  be an  $n$ -ary randomized function. Let  $\Pi$  be an  $n$ -party protocol for computing  $\hat{f}$ . For a coalition  $T \subseteq \{1, \dots, n\}$ , the view of  $T$  during an execution of  $\Pi$  on  $\mathbf{x} = (x_1 \dots x_n)$ , denoted  $\text{View}_T(\mathbf{x})$ , is defined as in Definition 2.4, i.e.,  $\text{View}_T(x_1, \dots, x_n)$  is the random variable containing the inputs of the parties in  $T$  (that is,  $\{x_i\}_{i \in T}$ ), the random inputs of the parties in  $T$ , and the messages that the parties in  $T$  received during the execution of the protocol with inputs  $\mathbf{x} = (x_1, \dots, x_n)$ .*

*We say that  $\Pi$  is a  $t$ -secure protocol for  $\hat{f}$  if there exists a randomized function, denoted  $S$ , such that for every  $t' \leq t$ , for every coalition  $T = \{i_1, \dots, i_{t'}\}$ , and for every inputs  $\mathbf{x} = (x_1 \dots x_n)$ , the following two random variables are identically distributed:*

- $\{S(T, (x_{i_1}, \dots, x_{i_{t'}}), o), o\}$ , where  $o$  is obtained first by sampling  $\hat{f}(\mathbf{x})$  (recall that  $\hat{f}$  is a randomized function) and then  $S$  is applied to  $(T, (x_{i_1}, \dots, x_{i_{t'}}), o)$ .
- $\{\text{View}_T(\mathbf{x}), \text{Output}^\Pi(\text{View}_T(\mathbf{x}))\}$ , where  $\text{Output}^\Pi(\text{View}_T(\mathbf{x}))$  denotes the output during the execution represented in  $\text{View}_T(\mathbf{x})$ .

**Claim 6.2** *Let  $\mathbf{y}$  and  $\mathbf{z}$  be two inputs and  $T$  be a coalition of size at most  $t$  such that  $\hat{f}(\mathbf{y})$  and  $\hat{f}(\mathbf{z})$  are identically distributed and  $y_i = z_i$  for every  $i \in T$ . In every  $t$ -secure protocol for  $\hat{f}$ , for any possible view  $v_T$  of the set  $T$ , it holds that  $\Pr[\text{View}_T(\mathbf{y}) = v_T] = \Pr[\text{View}_T(\mathbf{z}) = v_T]$ .*

**Proof:** Let  $T = \{i_1, \dots, i_{t'}\}$  for  $t' \leq t$ . The two random variables  $\{S(T, (y_{i_1}, \dots, y_{i_{t'}}), o), o\}$  and  $\{S(T, (z_{i_1}, \dots, z_{i_{t'}}), o), o\}$  (as defined in Definition 6.1) are identically distributed since  $\hat{f}(\mathbf{y})$  and  $\hat{f}(\mathbf{z})$  are identically distributed. Hence, by the  $t$ -security of the protocol, so do  $\{\text{View}_T(\mathbf{y}), \text{Output}^\Pi(\mathbf{y})\}$  and  $\{\text{View}_T(\mathbf{z}), \text{Output}^\Pi(\mathbf{z})\}$ .  $\square$

**Definition 6.3 (Symmetric Randomized Function)** We say that a randomized function  $\hat{f}$  over domain  $D$  with range  $R$  is symmetric if it does not depend on the ordering on the coordinates of the input, i.e., for every  $(x_1, \dots, x_n) \in D^n$  and every permutation  $\pi : [n] \rightarrow [n]$  the distributions (over  $R$ ) implied by  $\hat{f}(x_1, \dots, x_n)$  and by  $\hat{f}(x_{\pi(1)}, \dots, x_{\pi(n)})$  are identical.

Note that allowing  $O(nt)$  messages, it is fairly straightforward to construct a symmetric  $(t, \epsilon)$ -differentially private protocol with constant  $(O(1/\epsilon))$  additive error for  $\text{SUM}_n$ , using the natural paradigm with, say, the  $\epsilon$ -private approximation described in Example 2.3. The following lemma shows that  $\Omega(nt)$  messages are essential whenever a symmetric approximation for  $\text{SUM}_n$  is computed by an SFE protocol, even if it is not required to preserve differential privacy.

**Lemma 6.4** Let  $\hat{f}$  be a symmetric randomized function approximating  $\text{SUM}_n$  such that for every input vector  $\mathbf{x}$ , it holds that  $\Pr \left[ \left| \hat{f}(\mathbf{x}) - \text{SUM}(\mathbf{x}) \right| < n/4 \right] < 1/2$ , and let  $t \leq n - 2$ . Every fixed-communication  $t$ -secure protocol  $\Pi$  for computing  $\hat{f}$  uses at least  $n(t + 1)/4$  messages<sup>11</sup>.

**Proof:** Let  $\Pi$  be a  $t$ -secure protocol computing  $\hat{f}$  using less than  $n(t + 1)/4$  messages. Then, there are at least  $n/2$  lonely parties in  $\Pi$ . The intuition for the proof is that a lonely party does not affect the computation, since its neighboring set, being smaller than  $t + 1$ , would otherwise be able to infer information about its input. The proof is given in two steps. In the first step, we show that for any given lonely party  $p_i$ , for any fixed inputs for all other parties, and for any transcript  $c$  of the protocol, the probability of  $c$  being the transcript of the protocol when  $x_i = 0$  is exactly the same as the probability of  $c$  being the transcript of the protocol when  $x_i = 1$ . In the second step of the proof, we use this to show that with probability at least  $1/2$ , the protocol errs by  $n/4$ .

Without loss of generality, assume  $p_1$  is lonely and assume  $p_2$  is not a neighbor of  $p_1$ . Let  $T$  be the set of  $p_1$ 's neighbors and let  $R = \{p_1, \dots, p_n\} \setminus (T \cup \{p_1\})$  (in particular,  $p_2 \in R$ ). Recall that for a transcript  $c$  we denote by  $\alpha_1^c(x_1)$ , the probability that  $p_1$  is consistent with  $c$  with input  $x_1$ , namely, the probability that  $p_1$  with input  $x_1$  sends at each round messages according to  $c$ , provided it received all messages according to  $c$  in previous rounds. Our goal in the first part of the proof is to prove that for any transcript of the protocol  $c$ , it holds that  $\alpha_1^c(0) = \alpha_1^c(1)$ . Toward this end, we pursue the following proof structure.

- We first consider two inputs  $\mathbf{z}$  and  $\mathbf{y}$  such that  $\text{SUM}(\mathbf{z}) = \text{SUM}(\mathbf{y})$ ,  $y_i = z_i$  for every  $i \in T$ , but  $y_1 = 0$  while  $z_1 = 1$ . For every communication  $c$  exchanged in  $\Pi$ , denote  $c_T$  to be the messages sent and received by the parties in  $T$ . By Claim 6.2, since  $\hat{f}$  is symmetric, the probability of  $c_T$  is the same with  $\mathbf{z}$  and with  $\mathbf{y}$ .
- We simulate the protocol  $\Pi$  by a three-party protocol  $\Pi'$ , where the parties are  $p_1$ ,  $T$ , and  $R$ , and each one of them simulates the respective set of parties in  $\Pi$ . We then use Lemma 2.10 to write the probability that  $c_T$  is the communication exchanged in  $\Pi$  as a product of  $\alpha_1^{c_T}(x_1)$ ,  $\alpha_T^{c_T}(\mathbf{x}_T)$ , and

<sup>11</sup>We note that the lemma does not hold for non-symmetric functions. For example, we can modify the bit flip protocol described in Section 3 to an SFE protocol for a non-symmetric function, retaining the number of messages sent (but not their length): in Step (2)  $p_1$  also sends  $\mathbf{z} = (z_1, \dots, z_n)$ , and in Step (3) each  $p_i$  locally outputs  $\hat{f} + \mathbf{z}2^{-n}$ , treating  $\mathbf{z}$  as an  $n$ -bit binary number.

$\alpha_R^{c_T}(\mathbf{x}_R)$ , where  $\mathbf{x}_T$  (respectively,  $\mathbf{x}_R$ ) are the inputs of parties in  $T$  (respectively, in  $R$ ). We conclude that

$$\alpha_1^{c_T}(y_1) \cdot \alpha_T^{c_T}(\mathbf{y}_T) \cdot \alpha_R^{c_T}(\mathbf{y}_R) = \alpha_1^{c_T}(z_1) \cdot \alpha_T^{c_T}(\mathbf{z}_T) \cdot \alpha_R^{c_T}(\mathbf{z}_R).$$

Furthermore,  $\alpha_T^{c_T}(\mathbf{y}_T) = \alpha_T^{c_T}(\mathbf{z}_T)$  (since  $\mathbf{y}_T = \mathbf{z}_T$ ), thus,

$$\alpha_1^{c_T}(y_1) \cdot \alpha_R^{c_T}(\mathbf{y}_R) = \alpha_1^{c_T}(z_1) \cdot \alpha_R^{c_T}(\mathbf{z}_R).$$

- We then assert, by considering all prefixes of  $c_T$ , that each factor of these two multiplications is the same in both cases and hence  $\alpha_1^c(0) = \alpha_1^{c_T}(0) = \alpha_1^{c_T}(1) = \alpha_1^c(1)$ .

**Formal proof.** Fix any inputs  $x_3, \dots, x_n$  for the parties  $p_3, \dots, p_n$ . Let  $\mathbf{y}$  be the input vector

$$y_1 = 0, y_2 = 1, \text{ and } y_k = x_k \text{ for } 3 \leq k \leq n,$$

and let  $\mathbf{z}$  be the input vector

$$z_1 = 1, z_2 = 0, \text{ and } z_k = x_k \text{ for } 3 \leq k \leq n.$$

We first claim that the distribution over the views of  $T$  when the protocol is executed with  $\mathbf{y}$  is the same as when the protocol is executed with  $\mathbf{z}$ . As  $\text{SUM}(\mathbf{y}) = \text{SUM}(\mathbf{z})$  and  $\hat{f}$  is symmetric,  $\hat{f}(\mathbf{y})$  and  $\hat{f}(\mathbf{z})$  are identically distributed. Hence, by Claim 6.2, for any possible view  $v_T$  of the set  $T$ , it holds that  $\Pr[\text{View}_T(\mathbf{y}) = v_T] = \Pr[\text{View}_T(\mathbf{z}) = v_T]$ . Thus, since the view of  $T$  contains the transcript  $c_T$  of messages sent between the parties in  $T$  and the parties in  $\{p_1\} \cup R$ , we have that for any such possible transcript  $c_T$ , the probability that the parties send messages according to  $c_T$  is the same when the protocol is executed with  $\mathbf{y}$  and when the protocol is executed with  $\mathbf{z}$ . Furthermore, for any possible prefix  $c'_T$  of any transcript  $c_T$  of  $T$ , the probability of messages sent according to  $c'_T$  when executing  $\Pi$  with input  $\mathbf{y}$  is the same as when executing  $\Pi$  with input  $\mathbf{z}$ . This is true as this probability is merely the sum over the probabilities of all transcripts completing  $c'_T$ .

Without loss of generality, we can analyze the execution of the protocol as if at each round only a single message is sent by a single party. Let  $j$  be such that  $p_1$  sends a message in round  $j$  and denote by  $h_j = h_{j-1}, m_j$ , the prefix of  $c_T$  also viewed by  $p_1$  (messages sent or received by  $p_1$ ) in the first  $j$  rounds, where  $h_{j-1}$  is the history of messages viewed by  $p_1$  in the first  $j-1$  rounds, and  $m_j$  is the message  $p_1$  sends in round  $j$ , according to  $c_T$ . By the argument above, the probabilities of  $h_{j-1}$  being seen by  $p_1$  are the same when the protocol is executed with  $\mathbf{y}$  and when the protocol is executed with  $\mathbf{z}$  and the probabilities of  $h_j$  being seen by  $p_1$  are the same when the protocol is executed with  $\mathbf{y}$  and when the protocol is executed with  $\mathbf{z}$ . Thus, the probabilities of  $p_1$  sending  $m_j$  having seen message history  $h_{j-1}$  are the same when  $x_1 = 0$  and when  $x_1 = 1$ . Since the probability of  $p_1$  being consistent with a view  $c_T$  (of  $T$ ) is the product of the probabilities that it is consistent at each round, we have  $\alpha_1^{c_T}(0) = \alpha_1^{c_T}(1)$ . Let  $c$  be a full transcript of the protocol, and  $c_T$  be its restriction to messages sent between parties in  $T$  and parties in  $\{p_1\} \cup R$ . Since  $p_1$  does not see any message in  $c$  that is not in  $c_T$ , it holds for every  $x_1$  that  $\alpha_1^c(x_1) = \alpha_1^{c_T}(x_1)$ . Thus,  $\alpha_1^c(0) = \alpha_1^c(1)$ .

Hence, we proved that for any lonely party  $p_i$ , and any full transcript of the protocol  $c$ , it holds that  $\alpha_i^c(0) = \alpha_i^c(1)$ . Consider the all zero input vector and the input vector  $\mathbf{x}$  such that  $x_i = 1$  if and only if  $p_i$  is lonely. By Lemma 2.10 we have that for any given full transcript  $c$ , the probability of  $c$  being exchanged with  $\mathbf{0}$  is exactly the probability of  $c$  being exchanged with  $\mathbf{x}$ . Thus, if with probability at least  $1/2$ , when executing the protocol with  $\mathbf{0}$ , the exchanged transcript implies a value less than  $n/4$ , then with probability at least  $1/2$ , the protocol errs by at least  $n/4$  when executed with  $\mathbf{x}$ . Otherwise, with probability at least  $1/2$ , the protocol errs by at least  $n/4$  when executed with  $\mathbf{0}$ .  $\square$

**Acknowledgments** We thank Adam Smith, Yuval Ishai, and Cynthia Dwork for conversations related to the topic of this paper. This research is partially supported by the Frankel Center for Computer Science, and by the Israel Science Foundation (grant No. 860/06).

## References

- [1] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for noncryptographic fault-tolerant distributed computations. In *Proc. of the 20th ACM Symp. on the Theory of Computing*, pages 1–10, 1988.
- [2] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the SuLQ framework. In *Proc. of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138, 2005.
- [3] A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In *Proc. of the 40th ACM Symp. on the Theory of Computing*, pages 609–618, 2008.
- [4] D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols. In *Proc. of the 20th ACM Symp. on the Theory of Computing*, pages 11–19, 1988.
- [5] B. Chor and E. Kushilevitz. A communication-privacy tradeoff for modular addition. *Inform. Process. Lett.*, 45(4):205–210, 1993.
- [6] I. Damgård, M. Fitzzi, E. Kiltz, J. B. Nielsen, and T. Toft. Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation. In S. Halevi and T. Rabin, editors, *Proc. of the Third Theory of Cryptography Conference – TCC 2006*, volume 3876 of *Lecture Notes in Computer Science*, pages 285–304. Springer-Verlag, 2006.
- [7] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 202–210, 2003.
- [8] C. Dwork. Differential privacy. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Proc. of the 33rd International Colloquium on Automata, Languages and Programming*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer-Verlag, 2006.
- [9] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology – EUROCRYPT 2006*, pages 486–503, 2006.
- [10] C. Dwork and J. Lei. Differential privacy and robust statistics. In *Proc. of the 41st ACM Symp. on the Theory of Computing*, 2009.
- [11] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In S. Halevi and T. Rabin, editors, *Proc. of the Third Theory of Cryptography Conference – TCC 2006*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer-Verlag, 2006.
- [12] C. Dwork, F. McSherry, and K. Talwar. The price of privacy and the limits of LP decoding. In *Proc. of the 39th ACM Symp. on the Theory of Computing*, pages 85–94, 2007.

- [13] C. Dwork, M. Naor, O. Reingold, G. Rothblum, and S. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proc. of the 41st ACM Symp. on the Theory of Computing*, pages 381–390, 2009.
- [14] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In M. Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 528–544. Springer-Verlag, 2004.
- [15] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 211–222, 2003.
- [16] D. Feldman, A. Fiat, H. Kaplan, and K. Nissim. Private coresets. In *Proc. of the 41st ACM Symp. on the Theory of Computing*, 2009.
- [17] A. Ghosh, T. Roughgarden, and M. Sundararajan. Universally utility-maximizing privacy mechanisms. In *Proc. of the 41st ACM Symp. on the Theory of Computing*, pages 351–360, 2009.
- [18] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proc. of the 19th ACM Symp. on the Theory of Computing*, pages 218–229, 1987.
- [19] S. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *Proc. of the 49th IEEE Symp. on Foundations of Computer Science*, pages 531–540, 2008.
- [20] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Proc. of the 48th IEEE Symp. on Foundations of Computer Science*, pages 94–103, 2007.
- [21] I. Mironov, O. Pandey, O. Reingold, and S. P. Vadhan. Computational differential privacy. In S. Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 126–142. Springer-Verlag, 2009.
- [22] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *Proc. of the 39th ACM Symp. on the Theory of Computing*, pages 75–84, 2007.
- [23] V. Rastogi, S. Hong, and D. Suciu. The boundary between privacy and utility in data publishing. In *Proc. of the 33rd International Conf. on Very Large Data Bases*, pages 531–542, 2007.
- [24] A. Smith. Efficient, differentially private point estimators. Technical Report 0809.4794, CoRR, 2008.
- [25] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.
- [26] A. C. Yao. Protocols for secure computations. In *Proc. of the 23th IEEE Symp. on Foundations of Computer Science*, pages 160–164, 1982.