Distributed Private Data Analysis: Simultaneously Solving How and What

Amos Beimel, Kobbi Nissim, and Eran Omri

Department of Computer Science, Ben Gurion University, Be'er Sheva, Israel.
{beimel|kobbi|omrier}@cs.bgu.ac.il

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

1 Introduction

We examine the combination of two directions in the field of privacy concerning distributed private inputs – secure function evaluation [18, 13, 3, 1] and differential privacy [9, 7]. 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 $p_1, \ldots, p_n$, sharing a common interest in distributively computing a function $f(\cdot)$ of their inputs $\mathbf{x} = (x_i, \ldots, x_n)$, to compute $f(\mathbf{x})$ while making sure that no coalition of t or less curious parties learns anymore 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 [18, 13, 3, 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 [6, 12, 11, 2, 9, 7, 8]. Alongside, techniques have been developed for constructing a differentially private analysis $\hat{f}(\cdot)$ approximating a desired analysis $f(\cdot)$, by means of adding carefully chosen random noise that conceals any single individual's contribution to $f(\cdot)$ [9, 2, 16, 15].

Combining these two lines of research – SFE and differential privacy – we get a very natural paradigm for constructing protocols that preserve differential privacy, making use of the generality of SFE:

1. Decide on *what* to compute, i.e., a differentially private analysis $\hat{f}(\cdot)$ that approximates a desired analysis $f(\cdot)$. This 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 $\hat{f}(\cdot)$, 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. However, comparing the privacy requirements from SFE protocols with differential privacy suggests 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 function's outcome. Hence, it is interesting to learn whether there are advantages to a paradigm where the analysis to be computed and the protocol for computing it are chosen simultaneously.

The main model of distribution we consider is of semi-honest parties $p_1, \ldots, p_n$ that perform a computation over their private inputs $x_1, \ldots, x_n$, while maintaining differential privacy with respect to coalitions of size up to t (see Definition 2 below). This model has been examined thoroughly in cryptography, and was shown to enable SFE in a variety of settings [18, 13, 1, 3]. We note that while it is probably most natural to consider a setting where the players are computationally limited, we present our results in an information theoretic setting, as this setting allows us to prove lowerbounds on protocols, and hence demonstrate rigorously when constructing differentially private protocols is better than using the natural paradigm.

The second model we consider is the *local model*[1]. Protocols executing in the local model have a very simple communication structure, where each party p_i can only communicate with a designated semi-honest party C, referred to as a *curator*. The communication can either be *non-interactive*, where each party

[1] Also refereed to in the literature as *randomized response* and *input perturbation*. This model was originally introduced by Warner [17] to encourage survey responders to answer truthfully, and has been studied extensively since.

sends a single message to the curator which replies with the protocol's outcome, or *interactive*, where several rounds of communication may take place.

1.1 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. The latter example is of the Binary Sum function, $\mathrm{SUM}(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i$ for $x_i \in \{0,1\}$.

The major part of this work examines whether constructing protocols for computing an approximation $\hat{f}(\cdot)$ to $\mathrm{SUM}(\cdot)$, that are not SFE protocols for $\hat{f}(\cdot)$, yields an efficiency gain[2]. 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. We prove first that no such local non-interactive protocols exist (by itself, this contribution is not new, see below). Furthermore, no local protocols with $\ell \leq \sqrt{n}$ rounds and additive error $\sqrt{n}/\tilde{O}(\ell)$ exist. In particular, no local interactive protocol with $o(\sqrt{n/\log(n)})$ rounds exists for computing $\mathrm{SUM}(\cdot)$ within constant additive error[3]. Finally, the bounds on local protocols imply that no distributed protocols exist that use $o(nt)$ messages, and approximates $\mathrm{SUM}(\cdot)$ within additive error $\sqrt{n}/\tilde{O}(\ell)$ in ℓ rounds. Considering the natural paradigm, i.e., computing a differentially-private approximation to $\mathrm{SUM}(\cdot)$ using SFE, we get a protocol for approximating $\mathrm{SUM}(\cdot)$ with $O(1)$ additive error, and sending $O(nt)$ messages.

1.2 Techniques

We prove our lowerbound in sequence of reductions. We begin with a simple reduction from any differentially private protocol for SUM to a gap version of the threshold function 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

[2] We only consider *oblivious protocols* where the communication pattern is independent of input and randomness (see Section 2.2).

[3] This is in contrast to the centralized setup where $\mathrm{SUM}(\cdot)$ can be computed within $O(1)$ additive error.

distributed differentially-private protocols for GAP-TR($\cdot$) in which half of the parties interact with less than $t+1$ other parties, then there exist differentially-private protocol for GAP-TR($\cdot$) in the local interactive model. This allows us to prove our impossibility results in the local model, a model which is considerably simpler to analyze.

In analyzing the local non-interactive model, we prove lowerbounds borrowing from analyses in [6, 11]. The main technical difference is that our analysis holds for general protocols, whereas the work in [6, 11] was concerned with proving feasibility of privacy preserving computations, and hence the analysis of very specific protocols.

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

1.3 Related Work

Secure function evaluation and private data analysis were first tied together in the *Our Data, Ourselves (ODO)* protocols [8]. Their constructions – distributed SFE protocols for generating shares of random noise used in private data analyses – follow the natural paradigm discussed above. They do, however, avoid utilizing generic SFE feasibility results to gain on efficiency. We note that a point of difference between the protocols in [8] and the discussion herein is that ODO protocols are secure against malicious parties, in a computational setup, whereas we deal with semi-honest parties in an information theoretic setup.

Lowerbounds on the local non-interactive model were previously presented implicitly in [9, 14], and explicitly in [6, 10]. 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 $o(\sqrt{n})$, a similar consequence to our analysis of local non-interactive protocols. However (to the best of our understanding), these implications of [6, 10] do not imply the lowerbounds we get for local interactive protocols.

Chor and Kushilevitz [4] 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 also only prove lowerbounds for oblivious protocols.

2 Preliminaries

Notation. D denotes an arbitrary domain. A *vector* $\mathbf{x} = (x_1, \dots, x_n)$ is an ordered sequence of n elements of 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*, $\mathrm{Lap}(\lambda)$, is the continuous probability distribution with probability density function $h(y) = \frac{\exp(-|y|/\lambda)}{2\lambda}$ (hence, $\mathbb{E}[Y] = 0$, $\mathrm{Var}[Y] = 2\lambda^2$, and $\Pr[|Y| > k\lambda] = e^{-k}$).

2.1 Differential Privacy

Our privacy definition (Definition 2 below) can be viewed as a distributed variant of ϵ-differential privacy (a.k.a. ϵ-indistinguishability). Informally, a computation is differentially private if any change in a single private input may only induce a small change in the distribution on its outcomes.

Definition 1 ([9]). *Let $\hat{f} : \mathcal{D}^n \to R$ be a randomized function from domain D^n to range R. We say that $\hat{f}(\cdot)$ is* ϵ-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* $\Pr[\hat{f}(\mathbf{x}) \in \mathcal{V}] \leq e^{\epsilon} \cdot \Pr[\hat{f}(\mathbf{x}') \in \mathcal{V}]$. *The probability is taken over the randomness of $\hat{f}(\cdot)$.*

Several frameworks for constructing differentially private functions by means of perturbation are presented in the literature (see [9, 2, 16, 15]). The most basic transformation on a function f that yields a differentially private function is via the framework of *global sensitivity* [9], where the outcome $f(\mathbf{x})$ is modified by the addition of noise sampled from the Laplace distribution, calibrated to the global sensitivity of f,

$$\hat{f}(\mathbf{x}) = f(\mathbf{x}) + Y, \tag{1}$$

where $Y \sim \mathrm{Lap}(\mathrm{GS}_f/\epsilon)$, and $\mathrm{GS}_f = \max |f(\mathbf{x}) - f(\mathbf{x}')|$, with the maximum taken over neighboring $\mathbf{x}, \mathbf{x}'$.

Example 1. The binary sum function $\mathrm{SUM} : \{0,1\}^n \to \mathbb{R}$ is defined as $\mathrm{SUM}(\mathbf{x}) = \sum_{i=1}^{n} x_i$. For every two neighboring $\mathbf{x}, \mathbf{x}' \in \{0,1\}^n$ we have that $|\mathrm{SUM}(\mathbf{x}) - \mathrm{SUM}(\mathbf{x}')| = 1$ and hence $\mathrm{GS}_{\mathrm{SUM}} = 1$. Applying (1), we get an ϵ-differentially private approximation, $\hat{f}(\mathbf{x}) = \mathrm{SUM}(\mathbf{x}) + Y$, where $Y \sim \mathrm{Lap}(1/\epsilon)$.

2.2 Differentially Private Protocols

We consider a distributed setting, where semi-honest parties $p_1, \ldots, p_n$ hold private inputs $x_1, \ldots, x_n$ respectively and engage in a protocol Π in order to compute (or approximate) a function $f(\cdot)$ of their joint inputs. The protocol Π 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 size up to t. Following [4], we only consider a *fixed-communication* protocol Π (also called an oblivious protocol) where every channel is either (i) active in every run of Π (i.e., at least one bit is sent over the channel), or (ii) never used[4]. Parties that are adjacent to at least $t+1$ active channels are called *popular* other parties are called *lonely*.

[4] 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 Π (where the probability is taken over the coins of Π), or (ii) is never used.

The main definition we will work with is an extension of Definition 1 to a distributed setting. Informally, we require that differential privacy is preserved with respect to any coalition of size up to t.

Definition 2. *Let Π be a protocol between n (semi-honest) parties. For a set $T \subseteq [n]$, let $\mathrm{View}_T(x_1, \ldots, x_n)$ be the random variable containing the inputs of the parties in T (i.e. $\{x_i\}_{i \in T}$), the random coins of the parties in T, and the messages that the parties in T received during the execution of the protocol with private inputs $\mathbf{x} = (x_1, \ldots, x_n)$.*

We say that Π is (t, ϵ)-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 parties in T:*

$$\Pr[\mathrm{View}_T(\mathbf{x}) \in \mathcal{V}_T] \leq e^{\epsilon} \cdot \Pr[\mathrm{View}_T(\mathbf{x}') \in \mathcal{V}_T], \tag{2}$$

where the probability is taken over the randomness of the protocol Π.

It is possible to relax this definition by replacing (2) by a requirement that View_T is statistically close, or computationally close to some ϵ-differentially private computation. The exact definition of differential privacy in the computational model requires some care; this definition will be given in the full version of the paper. The following informal lemma applies for such relaxations:

Lemma 1 (Informal). *Let $\hat{f}$ be ϵ-differentially private, and let Π be a t-secure protocol computing $\hat{f}$, then Π is (t, ϵ)-differentially private.*

Note 1. While a computational definition of differentially private protocols is probably the most appealing, we chose to present our work with Definition 2 – an *information theoretic* definition of differentially private protocols – because it allows us to prove bounds on protocols, demonstrating when constructing differentially private protocols is better than using the natural paradigm.

Note 2. 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 the $\hat{f}(\mathbf{x})$ and send the outcome to all other parties.

2.3 The Local Model

The local model (previously discussed in [9, 14]) is a simplified distributed communication model where the parties communicate via a designated party – a *curator* – denoted C (with no local input). We will consider two types of differentially private local protocols. In *non-interactive* local protocols each party p_i applies an ϵ-differentially private algorithm $\mathcal{A}_i$ on its private input x_i and randomness r_i, and sends $\mathcal{A}_i(x_i; r_i)$ to C that then performs an arbitrary computation and publishes its result.

In *interactive* local protocols the input to each algorithm $\mathcal{A}_i$ includes x_i, r_i, and the history of messages received from the curator. The protocol proceeds in iterations where in each iteration C sends to party p_i a "query" message $q_{i,j}$ and party p_i responds with $\mathcal{A}_i(x_i; q_{i,1}, \ldots, q_{i,j}; r_i)$. It is required that the overall protocol preserves differential privacy, i.e., that the randomized function corresponding to the curator's view $\mathcal{A}_i(\cdot; q_{i,1}; r_i) \circ \mathcal{A}_i(\cdot; q_{i,1}, q_{i,2}; r_i) \circ \cdots \circ \mathcal{A}_i(\cdot; q_{i,1}, \ldots, q_{i,j}; r_i)$ preserves ϵ-differential privacy for every query messages $q_{i,1}, \ldots, q_{i,j}$ possible in the protocol. An immediate corollary is that $\mathcal{A}_i(\cdot; q_{i,1}, \ldots, q_{i,j}; r_i)$ should be ϵ-differentially private for all j.

2.4 Approximation

We will construct protocols whose outcome approximates a function $f : D^n \to \mathbb{R}$ by a probabilistic function. We say that a randomized function $\hat{f} : D^n \to \mathbb{R}$ is an *additive* (γ, τ)-approximation of f if $\Pr\left[|f(\mathbf{x}) - \hat{f}(\mathbf{x})| > \tau(n)\right] < \gamma(n)$ for all $\mathbf{x} \in D^n$. For example, Equation (1) yields an additive $(O(1), O(\mathrm{GS}_f/\epsilon))$-approximation to f.

3 Motivating Examples

We begin with two observations 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.

Binary Sum – $\sqrt{n}$ Additive Error. We begin with a simple observation regarding the binary sum function of Example 1: a very efficient $(n-1, \epsilon)$-differentially private protocol for approximating $\mathrm{SUM}(\mathbf{x}) = \sum_{i=1}^{n} x_i$ (where $x_i \in \{0,1\}$) within $O(\sqrt{n}/\epsilon)$-additive approximation.

Let $\mathrm{flip}(x)$ be a randomized bit flipping operator returning x with probability $0.5 + \alpha$ and $1 - x$ otherwise (α will be determined later). Our protocol proceeds as follows: (i) Each party p_i with private input $x_i \in \{0,1\}$ sends $z_i = \mathrm{flip}(x_i)$ to party p_1; (ii) Party p_1 sends $k = \sum_{i=1}^{n} z_i$ to all parties; (iii) Each party p_i locally outputs $\hat{f} = (k + (0.5 - \alpha)n)/2\alpha$. In this protocol, a total of $O(n)$ messages and $O(n \log n)$ bits of communication are exchanged.

To satisfy Definition 2, set $\alpha = \frac{\epsilon}{4+2\epsilon}$, yielding $\Pr[z_i = x_i]/\Pr[z_i = 1 - x_i] = (0.5 + \alpha)/(0.5 - \alpha) = 1 + \epsilon \leq e^{\epsilon}$.

Note that $\mathbb{E}[k] = (0.5+\alpha)\,\mathrm{SUM}(\mathbf{x}) + (0.5-\alpha)(n - \mathrm{SUM}(\mathbf{x})) = 2\alpha\,\mathrm{SUM}(\mathbf{x}) + n(0.5-\alpha)$, and hence, $\mathbb{E}[\hat{f}] = \mathrm{SUM}(\mathbf{x})$. By an application of the Chernoff bound, we get that $\hat{f}$ is an additive $(O(1), O(\sqrt{n}/\epsilon))$-approximation to $\mathrm{SUM}(\cdot)$.

It is natural to choose a *symmetric* approximation to $\mathrm{SUM}(\cdot)$ that only depends on $\mathrm{SUM}(\cdot)$. While the construction above yields an efficient protocol for such a function, we prove (using ideas from [4]) that no efficient SFE protocols for such functions exist. We leave the details for the full version.

Lemma 2. *Let $\hat{f}$ be a symmetric additive $(O(1), n/10)$-approximation to* $\mathrm{SUM}(\cdot)$. *Then any oblivious t-secure protocol computing $\hat{f}$ uses $\Omega(nt)$ messages*[5].

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 subsequence of length n^α. In other words, the minimum weight over all substrings of $\mathbf{x}$ of length n^α bits: $\mathrm{DIST}_\alpha(\mathbf{x}) = \min_i(\sum_{j=i}^{i+n^\alpha-1} x_j)$. For $t \leq n/2$, we present a (t, ϵ, δ)-differentially private protocol[6] approximating $\mathrm{DIST}_\alpha(\mathbf{x})$ with additive error $\tilde{O}(n^{\alpha/3}/\epsilon)$.

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 $i_1, \ldots, i_{n^{1-\alpha/3}}$ 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 DIST_α. The protocol for computing an approximation $\hat{f}$ to DIST_α 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 $R = \frac{2\log(\frac{2}{\delta})}{\epsilon^2}$, and shares $x_i + Y_i$ between parties $p_1, \ldots, p_{t+1}$ using an additive $(t+1)$-out-of-$(t+1)$ secret sharing scheme.
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 [8], this protocol is a (t, ϵ, δ)-differentially private protocol when $2t < n$. Furthermore, it can be shown that with high probability the additive error is $\tilde{O}(n^{\alpha/3}/\epsilon)$. To conclude, we showed a simple 3 round protocol for DIST_α.

This protocol demonstrates two advantages of the paradigm of choosing what and how together. First, we choose an approximation of DIST_α (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.[7]

---

[5] We note that the lemma does not hold for non-symmetric functions. For example, we can modify the bit flip protocol above to an SFE protocol for a non-symmetric function, retaining the number of messages sent (but not their length): in step (iii) let p_1 send $\mathbf{z} = (z_1, \ldots, z_n)$, and in step (iv) let p_i locally output $\hat{f} + \mathbf{z}2^{-n}$, treating $\mathbf{z}$ as an n-bit binary number.

[6] (ϵ, δ)-differential privacy is a generalization, defined in [8], of ϵ-differential privacy where it is only required that $\Pr[\hat{f}(\mathbf{x}) \in \mathcal{V}] \leq e^\epsilon \cdot \Pr[\hat{f}(\mathbf{x}') \in \mathcal{V}] + \delta$.

[7] One can use the techniques of [5] to avoid leaking these sums while maintaining a constant number of rounds, however the resulting protocol is less efficient.

4 Binary Sum – Below $\sqrt{n}$ Additive Error

We prove that in any ℓ-round, fixed-communication, (t, ϵ)-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 $\Omega(nt)$.

Theorem 1. *In any ℓ-round, fixed-communication, (t, ϵ)-differentially private protocol for approximating* SUM_n *that sends at most $n(t+1)/4$ messages the error is $\Omega(\sqrt{n}/(\epsilon\ell\sqrt{\log \ell}))$ with constant probability.*

We prove this lowerbound in steps. We first define a gap version of the threshold function, denoted GAP-TR, and observe that any differentially private protocol for SUM with error τ implies a differentially-private protocol for GAP-TR with gap $\tau/2$. Therefore, we prove impossibility of differentially-private computation of GAP-TR with small gap. In Section 4.1, we prove that if there is a protocol computing the GAP-TR function with at most $nt/4$ messages, then there is a protocol in the local model (i.e., with a curator) computing the GAP-TR function with the same gap. Thereafter, we prove that such a protocol in the local model has can only compute GAP-TR with gap $\Omega(\sqrt{n})$. In Section 4.2, we analyze properties of non-interactive protocols in the local model that compute GAP-TR and in Section 4.3 we generalize this analysis to interactive protocols in the local model that compute GAP-TR. In Section 4.4, we complete the proof of the lowerbound on the gap in the local model. Theorem 1 follows from the combination the transformation of the distributed protocol to the protocol in local model proved in Lemma 4 and the lowerbound for protocols in the local model proved in Theorem 2.

Theorem 1 suggests that whenever we require that the error of a differentially-private protocol for approximating $\mathrm{SUM}(\cdot)$ to be of magnitude smaller than $\sqrt{n/\epsilon}$, 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 $(O(1/\epsilon), O(1))$-approximation of $\mathrm{SUM}(\cdot)$.

We next define the gap version of the threshold function:

Definition 3 (Gap Threshold Function). *We define the gap threshold function as follows: If* $\mathrm{SUM}_n(x_1, \ldots, x_n) \leq \kappa$ *then* $\mathrm{GAP\text{-}TR}_{\kappa,\tau}(x_1, \ldots, x_n) = 0$ *and if* $\mathrm{SUM}_n(x_1, \ldots, x_n) \geq \kappa + \tau$ *then* $\mathrm{GAP\text{-}TR}_{\kappa,\tau}(x_1, \ldots, x_n) = 1$.

In the above definition we consider a gap version of the threshold function and there are no requirements on the output of $\mathrm{GAP\text{-}TR}_{\kappa,\tau}$ when $\kappa < \mathrm{SUM}_n(x_1, \ldots, x_n) < \kappa + \tau$.

Claim. If there exists an ℓ-round, fixed-communication, (t, ϵ)-differentially private protocol that (γ, τ)-approximates SUM_n sending at most $n(t+1)/4$ messages, then for every κ there exists an ℓ-round, (t, ϵ)-differentially private protocol that correctly computes $\mathrm{GAP\text{-}TR}_{\kappa,\tau/2}$ with probability at least γ sending at most $n(t+1)/4$ messages.

Similarly, using "padding" arguments

Claim. If for some $0 \leq \kappa \leq n - \tau$ there exists an ℓ-round, fixed-communication, (t, ϵ)-differentially private n party protocol that correctly computes $\text{GAP-TR}_{\kappa,\tau}$ with probability at least γ sending at most $n(t+1)/4$ messages, then there exists an ℓ-round, (t, ϵ)-differentially private $n/2$-party protocol that correctly computes $\text{GAP-TR}_{0,\tau}$ with probability at least γ sending at most $n(t+1)/4$ messages.

4.1 Moving to the Local Model

We start with the transformation of a distributed protocol to a protocol in the local model. To analyze this transformation we will need the following simple lemma:

Lemma 3. *Fix a 3-party randomized protocol, assume that each p_i holds an inputs x_i, and fix some communication transcript c. Define α_i as the overall probability that in each round p_i with input x_i sends messages according to c provided that in previous rounds it gets messages according to c. Then, the probability that c is exchanged is $\alpha_1 \cdot \alpha_2 \cdot \alpha_3$.*

Lemma 4. *If there exists an ℓ-round, (t, ϵ)-differentially private protocol that correctly computes $\text{GAP-TR}_{\kappa,\tau}$ with probability at least γ sending at most $n(t+1)/4$ messages, then there exists a 2ℓ-round, ϵ-differentially private protocol in the local model that correctly computes $\text{GAP-TR}_{\kappa,\tau}$ with probability at least γ.*

Proof. Assume that there is a distributed protocol Π satisfying the conditions in the lemma. Recall that a party in Π is lonely if it has at most t neighbors and it is popular otherwise. 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 computes $\text{GAP-TR}_{\kappa,\tau}$ for $n/2$ parties in two stages: (i) We first construct a protocol $\mathcal{P}$ in the local model which computes $\text{GAP-TR}_{\kappa,\tau}$ for n parties and only protects the privacy of the lonely parties. (ii) We next 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 Stage. We convert the distributed protocol Π to a protocol $\mathcal{P}$ in the local model as follows: We have two rounds in $\mathcal{P}$ for every round of Π. For every message m that Party p_j sends to Party p_k in round i in Protocol Π, Party p_j sends m to the curator in round $2i-1$ and the curator sends m to Party p_k in round $2i$. 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, T be the set of size at most t containing the neighbors of p_1, and $R = \{p_1, \ldots, p_n\} \setminus (T \cup \{p_1\})$. 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,

$$\Pr[\text{View}_{\mathcal{C}}(\mathbf{x}) = v] \leq e^{\epsilon} \cdot \Pr[\text{View}_{\mathcal{C}}(\mathbf{x}') = v]$$

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

Fix a view v of the curator. For a set A, define α_A and α'_A as the probabilities in Π 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 $\alpha_A = \alpha'_A$. By simulating p_1, T, R by three parties and applying Lemma 3, and by the construction of $\mathcal{P}$ from Π

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

Thus, we need to prove that

$$\alpha_{\{p_1\}} \leq e^{\epsilon} \alpha'_{\{p_1\}}. \tag{3}$$

We use the (t, ϵ) privacy of protocol Π to prove (3). 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 β_A and β'_A as the probabilities that in Π 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 3,

$$\Pr[\mathrm{View}^{\Pi}_{T}(\mathbf{x}) = (\mathbf{x_T}, \mathbf{r_T}, v_T)] = \alpha_{\{p_1\}} \cdot \beta_R, \quad \text{and}$$
$$\Pr[\mathrm{View}^{\Pi}_{T}(\mathbf{x}') = (\mathbf{x_T}, \mathbf{r_T}, v_T)] = \beta'_{\{p_1\}} \cdot \beta'_R = \alpha'_{\{p_1\}} \cdot \beta_R.$$

(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 $\mathbf{r_T}$. Therefore, the (t, ϵ) differential privacy of Π implies (3), and, thus, that $\mathcal{P}$ is ϵ-differentially private with respect to inputs of lonely parties.

Second Stage. There are at least $n/2$ lonely parties in Π, thus, w.l.o.g., parties $p_1, \ldots, p_{n/2}$ are lonely. We construct a protocol $\mathcal{P}'$ for computing GAP-TR$_{\kappa,\tau}$ for $n/2$ parties 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 Party p_i with input x_i sends them in $\mathcal{P}$; and (ii) 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 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 $\mathcal{P}'$ correctly computes GAP-TR$_{\kappa,\tau}$ with probability at least γ since we fixed $x_i = 0$ for $i < n/2 \leq n$ and $\mathcal{P}'$ returns the same output distribution of Π, which correctly computes GAP-TR$_{\kappa,\tau}$ with probability at least γ. $\square$

By Lemma 4 it suffices to prove lowerbounds on the gap τ for protocols in the local model.

4.2 GAP-TR in the Non-Interactive Local Model

We consider the non-interactive local model where each party holds an input $x_i \in \{0,1\}$ and independently applies an algorithm A_i (also called a sanitizer) before sending the sanitized result c_i to the curator. We consider a differentially-private protocol computing $\text{GAP-TR}_{0,\tau}$ in this model and we wish to prove lowerbounds on τ. Notice that we take $\kappa = 0$, namely, we want to prove that the curator cannot distinguish between the all-zero input and inputs of weight at least τ (for small values of τ). More formally, we want to prove that if each A_i is ϵ-differentially private, then the curator errs with constant probability when computing $\text{GAP-TR}_{0,\tau}$ for $\tau = O(\sqrt{n})$. Towards this goal, we show that there are two inputs for which the curator sees similar distributions on the messages, thus, has to return similar answers. However, one input contains $\Omega(\sqrt{n})$ ones and the other is the all-zero input, and the algorithm errs on at least one of the inputs. We will prove the existence of such input with $\Omega(\sqrt{n})$ ones, by considering a distribution $\mathcal{A}$ on inputs and later proving that such input taken from the distribution $\mathcal{A}$ exists.

We note that in the local model randomness for the curator can be supplied by the parties and hence we assume, w.l.o.g., that the curator is deterministic. Thus, the curator, having received the sanitized input $\mathbf{c} = S(\mathbf{x}) \stackrel{\Delta}{=} (A_1(x_1), \ldots, A_n(x_n))$, applies a deterministic algorithm G to $\mathbf{c}$, where $G(\mathbf{c})$ is supposed to answer $\text{GAP-TR}_{\kappa,\tau}(x_1, \ldots, x_n)$ correctly.

Let $\alpha \stackrel{\Delta}{=} \frac{1}{\epsilon}\sqrt{\frac{d}{n}}$ for d to be determined later. We consider two distributions over which the input is chosen.

- Distribution $\mathcal{A}$: $x_i = 1$ with probability α, $x_i = 0$ with probability $(1 - \alpha)$ (the inputs of the different parties are chosen independently).
- Distribution $\mathcal{B}$: $x_i = 0$ with probability 1 (that is, $\mathcal{B}$ always chooses the all-zero input vector).

From here on, we use X to identify the random variable representing the input and X_i for its ith coordinate. When considering the random variable over $\mathcal{A}$ (respectively, $\mathcal{B}$), we use the notation $\Pr_{\mathcal{A}}[\cdot]$ (respectively, $\Pr_{\mathcal{B}}[\cdot]$). For a set D, we use the notation $\Pr_{\mathcal{A}}[D]$ (respectively, $\Pr_{\mathcal{B}}[D]$) to denote the probability of the event that $A_i(X_i) \in D$ when X_i is generated according to the probability distribution $\mathcal{A}$ (respectively, $\mathcal{B}$).

We denote for every possible output $\mathbf{c} = (c_1, \ldots, c_n)$ of S,

$$r(\mathbf{c}) \stackrel{\Delta}{=} \frac{\Pr_{\mathcal{A}}\left[S(X) = \mathbf{c}\right]}{\Pr_{\mathcal{B}}\left[S(X) = \mathbf{c}\right]} \quad \text{and} \quad r_i(c_i) \stackrel{\Delta}{=} \frac{\Pr_{\mathcal{A}}\left[A_i(X_i) = c_i\right]}{\Pr_{\mathcal{B}}\left[A_i(X_i) = c_i\right]}. \tag{4}$$

Define a random variable $\mathbf{C} = (C_1, \ldots, C_n)$ where $C_i = A_i(X_i)$ and X_i is chosen according to the distribution $\mathcal{A}$. We next bound $\Pr_{\mathcal{A}}[r(\mathbf{C}) > \delta]$.

Lemma 5. $\Pr_{\mathcal{A}}[r(\mathbf{C}) > \exp(\nu d)] < \exp\left(-(\nu - 8)^2 d/8\right)$ *for every* $\nu > 8$.

We prove Lemma 5 using the Hoeffding bound. Define the random variables $V_i \stackrel{\Delta}{=} \ln r_i(C_i)$. For every $\eta > 0$, we have that

$$\Pr_{\mathcal{A}}[r(\mathbf{C}) > \eta] = \Pr_{\mathcal{A}}\left[\prod_{i=1}^{n} r_i(C_i) > \eta\right] = \Pr_{\mathcal{A}}\left[\sum_{i=1}^{n} V_i > \ln \eta\right], \tag{5}$$

where the first equality holds since the X_is are chosen independently. To apply the Hoeffding bound, we need to compute bounds on each variable V_i, and to compute the expectation of V_i. Both tasks are achieved using the ϵ-differential privacy of the sanitizers, that is,

$$e^{-\epsilon} \leq \frac{\Pr[A_i(1) = c_i]}{\Pr[A_i(0) = c_i]} \leq e^{\epsilon}. \tag{6}$$

Lemma 6. *$-2\alpha\epsilon \leq V_i \leq 2\alpha\epsilon$ for every i.*

Proof. For every i and every value c_i,

$$\begin{aligned} r_i(c_i) &= \frac{\alpha \Pr[A_i(1) = c_i] + (1-\alpha)\Pr[A_i(0) = c_i]}{\Pr[A_i(0) = c_i]} \\ &= 1 + \alpha \frac{\Pr[A_i(1) = c_i] - \Pr[A_i(0) = c_i]}{\Pr[A_i(0) = c_i]}. \end{aligned}$$

Using $\Pr[A_i(1) = c_i] \leq e^{\epsilon} \Pr[A_i(0) = c_i]$ we get on one hand that

$$r_i(c_i) \leq 1 + \alpha \frac{\Pr[A_i(0) = c_i]e^{\epsilon} - \Pr[A_i(0) = c_i]}{\Pr[A_i(0) = c_i]} \;=\; 1 + \alpha(e^{\epsilon} - 1) \;\leq\; 1 + 2\alpha\epsilon$$

(since $e^{\epsilon} < 1 + 2\epsilon$ for every $0 < \epsilon \leq 1$). Thus, $V_i = \ln r_i(C_i) \leq \ln(1 + 2\alpha\epsilon) \leq 2\alpha\epsilon$, since $\ln(1+x) \leq x$ for every $0 \leq x \leq 1$. Using $e^{-\epsilon} \Pr[A_i(0) = c_i] \leq \Pr[A_i(1) = c_i]$ we get on the other hand that

$$r_i(c_i) \geq 1 + \alpha \frac{\Pr[A_i(0) = c_i]e^{-\epsilon} - \Pr[A_i(0) = c_i]}{\Pr[A_i(0) = c_i]} \;=\; 1 + \alpha(e^{-\epsilon} - 1) \geq\; 1 - \alpha\epsilon.$$

Thus, $V_i = \ln r_i(C_i) \geq \ln(1 - \alpha\epsilon) \geq -2\alpha\epsilon$, since $\ln(1-x) \geq -2x$ for every $0 \leq x \leq 0.5$. □

Lemma 7. $\mathbb{E}[V_i] \leq 8\alpha^2\epsilon^2$.

Proof. In this proof we assume that the output of A_i is a countable set. Denote $B_b \stackrel{\Delta}{=} \{c_i : r_i(c_i) = 1 + b\}$ for every $-\alpha\epsilon \leq b \leq 2\alpha\epsilon$ (by Lemma 6, these are the only values possible for b). Note that by the definition of r_i, for every $c_i \in B_b$ $\Pr_{\mathcal{A}}[A_i(X_i) = c_i]/\Pr_{\mathcal{B}}[A_i(X_i) = c_i] = 1 + b$, thus, $\Pr_{\mathcal{B}}[B_b] = \frac{\Pr_{\mathcal{A}}[B_b]}{1+b} \leq (1 - b +$

$2b^2)\Pr_{\mathcal{A}}[B_b]$. Let $\beta = \alpha\epsilon$. We next bound $\mathbb{E}[V_i]$.

$$\begin{aligned}
\mathbb{E}[V_i] = \mathbb{E}_{\mathcal{A}}[\ln r(C_i)] &= \sum_{-\beta\le b\le 2\beta} \Pr_{\mathcal{A}}[B_b]\ln(1+b) \;\le\; \sum_{-\beta\le b\le 2\beta} \Pr_{\mathcal{A}}[B_b] b \\
&= \sum_{-\beta\le b\le 2\beta} \Pr_{\mathcal{A}}[B_b] - \sum_{-\beta\le b\le 2\beta} \Pr_{\mathcal{A}}[B_b](1-b+2b^2) + \sum_{-\beta\le b\le 2\beta} \Pr_{\mathcal{A}}[B_b](2b^2) \\
&\le \sum_{-\beta\le b\le 2\beta} \Pr_{\mathcal{A}}[B_b] - \sum_{-\beta\le b\le 2\beta} \Pr_{\mathcal{B}}[B_b] + \sum_{-\beta\le b\le 2\beta} \Pr_{\mathcal{A}}[B_b](2b^2) \\
&\le 1 - 1 + 8\beta^2 \sum_{-\beta\le b\le 2\beta} \Pr_{\mathcal{A}}[B_b] \;=\; 8\beta^2 = 8\alpha^2\epsilon^2. \qquad \square
\end{aligned}$$

From Lemma 7, $\mathbb{E}(\sum_{i=1}^{n} V_i) \le 8\alpha^2\epsilon^2 n = 8d$. We next prove Lemma 5 which shows that $\sum_{i=1}^{n} V_i$ is concentrated around this value.

Proof (of Lemma 5). We apply the Hoeffding bound: Let $V_1, \ldots, V_n$ be independent random variables such that $V_i \in [a, b]$. Then, $\Pr\left[\sum_{i=1}^{n} V_i - \mu \ge t\right] \le \exp\left(-\frac{2\,t^2}{n(b-a)^2}\right)$ for every $t > 0$ (where $\mu = \sum_{i=1}^{n} \mathbb{E}[X_i]$).

By (5), Lemma 6, and Lemma 7:

$$\begin{aligned}
\Pr_{\mathcal{A}}[r(\mathbf{C}) > \exp(\nu d)] &= \Pr_{\mathcal{A}}\left[\sum_{i=1}^{n} V_i > \nu d\right] = \Pr_{\mathcal{A}}\left[\sum_{i=1}^{n} V_i - \sum_{i=1}^{n}\mathbb{E}V_i > \nu d - \sum_{i=1}^{n}\mathbb{E}V_i\right] \\
&\le \Pr_{\mathcal{A}}\left[\sum_{i=1}^{n} V_i - \sum_{i=1}^{n}\mathbb{E}V_i > \nu d - n\cdot 8\alpha^2\epsilon^2\right] \\
&\le \exp\left(-\frac{2\,(\nu d - n\cdot 8\alpha^2\epsilon^2)^2}{16\,n\,\alpha^2\,\epsilon^2}\right) \;<\; \exp\left(-(\nu-8)^2 d/8\right) \qquad \square
\end{aligned}$$

The following corollary is a rephrasing of Lemma 5 that follows from the definition of r in (4) and the fact that distribution $\mathcal{B}$ picks the all-zero input with probability 1.

Corollary 1. *Assume we sample* $\mathbf{X}$ *according to distribution* $\mathcal{A}$ *and compute* $\mathbf{c} = S(\mathbf{X})$. *Then, for every* $\nu > 8$ *with probability at least* $1 - \exp\left(-(\nu-8)^2 d/8\right)$

$$\Pr_{\mathcal{A}}[S(\mathbf{Z}) = \mathbf{c}] \le \exp(-\nu d)\Pr[S(\mathbf{0^n}) = \mathbf{c}]\,,$$

where in the left hand side the probability is taken over the choice of Z *according to the distribution* $\mathcal{A}$ *and the randomness of the sanitizers and in the right hand side the probability is taken over the randomness of the sanitizers.*

4.3 GAP-TR$_{\kappa,\tau}$ in the Interactive Local Model

In this section we generalize Corollary 1 to interactive local protocols where each party holds an input $x_i \in \{0, 1\}$ and communicates with the curator in rounds. To achieve this goal, we decompose a 2ℓ-round $\epsilon/2$-differentially private protocol into ℓ protocols, and prove that each protocol is ϵ-differentially private. Thus, we can apply Corollary 1 to each protocol, and then apply a composition lemma.

Lemma 8. *Suppose we execute a ℓ-round, local, $\epsilon/2$-differentially private protocol and we sample a vector $\mathbf{X}$ according to distribution $\mathcal{A}$ and compute $\mathbf{c} = S(\mathbf{X})$ (where $S(\mathbf{X})$ is the communication in the 2ℓ rounds). Then, for every $\nu > 8$ with probability at least $1 - \ell \exp\left(-(\nu-8)^2 d/8\right)$*

$$\Pr_{\mathcal{A}}[S(\mathbf{Z}) = \mathbf{c}] \leq \exp\left(-\ell\nu d\right) \Pr[S(\mathbf{0^n}) = \mathbf{c}]\,,$$

where in the left side the probability is taken over the choice of Z according to the distribution $\mathcal{A}$ and the randomness of the sanitizers and in the right side the probability is taken over the randomness of the sanitizers.

Proof (Sketch). Fix a 2ℓ-round, $\frac{\epsilon}{2}$-differentially private, local protocol $\mathcal{P}$. In the interactive local model, a protocol is composed of ℓ-interactions where in each interaction the curator sends a query to each party and the party sends an answer.

Our first goal is to make the parties stateless. Consider a party p_i. First, we assume that in interaction j the curator sends all queries and answers $q_{i,1}, a_{i,1}, \ldots, a_{i,j-1}, q_{i,j}$ it sent and received from p_i in previous rounds. Second, we assume that 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 unbounded, such choice is possible). Party p_i uses this random string to answer the jth query. In other words, we can consider p_i as applying an algorithm $A_{i,j}$ to compute the jth answer; this algorithm depends on the previous queries and answers and uses an independent random string.

We next claim that $A_{i,j}$ is ϵ-differentially private. That is, we claim that the probability that $q_{i,j}$ 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. This follows from the following tow facts: (1) the probability of $q_{i,1}, a_{i,1}, \ldots, a_{i,j-1}, q_{i,j}$ is roughly the same when p_i holds the bit 0 and when p_i holds the bit 1. (2) the probability of $q_{i,1}, a_{i,1}, \ldots, a_{i,j-1}$ is roughly the same when p_i holds the bit 0 and when p_i holds the bit 1. The exact details are omitted. Thus, the answers of the n parties in interaction j are ϵ-private, and we can apply Corollary 1 to the concatenation of the n answers.

We now use the above protocol to construct a protocol $\mathcal{P}_1$ 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 (i.e., sends messages to the curator with the same distribution as the n parties send them). If the bit of the party in $\mathcal{P}_1$ is 1 it chooses the n input bits of the n parties in $\mathcal{P}$ according to distribution $\mathcal{A}$. If the bit of the party in $\mathcal{P}_1$ is 0 it chooses the n input bits of the n parties in $\mathcal{P}$ to be the all-zero vector. By Corollary 1 we can apply the composition lemma – Lemma 10 – to the composition of the ℓ non-interactive protocols and the lemma follows. □

Corollary 2. *For every $\nu > 8$ and for every set D of views in a 2ℓ-round protocol,*

$$\Pr_{\mathcal{B}}[D] \geq \frac{\Pr_{\mathcal{A}}[D] - \ell \exp\left(-(\nu-8)^2 d/8\right)}{\exp\left(\ell\nu d\right)}.$$

Proof. Let

$$D_1 = \left\{ \mathbf{c} \in D : \Pr_{\mathcal{A}}[S(X) = \mathbf{c}] \leq \exp\left(\ell\nu d\right) \Pr_{\mathcal{B}}[S(X) = \mathbf{c}] \right\}$$

and

$$D_2 = \left\{ \mathbf{c} \in D : \Pr_{\mathcal{A}}[S(X) = \mathbf{c}] > \exp\left(\ell\nu d\right) \Pr_{\mathcal{B}}[S(X) = \mathbf{c}] \right\}.$$

By Lemma 8, $\Pr_{\mathcal{A}}[D_2] \leq \ell \exp\left(-(\nu-8)^2 d/8\right)$. Furthermore, $\Pr_{\mathcal{B}}[D_1] \geq \frac{\Pr_{\mathcal{A}}[D_1]}{\exp(\ell\nu d)}$. Thus,

$$\Pr_{\mathcal{B}}[D] \geq \Pr_{\mathcal{B}}[D_1] \geq \frac{\Pr_{\mathcal{A}}[D_1]}{e^{\ell\nu d}} = \frac{\Pr_{\mathcal{A}}[D] - \Pr_{\mathcal{A}}[D_2]}{e^{\ell\nu d}} \geq \frac{\Pr_{\mathcal{A}}[D] - \ell e^{-(\nu-8)^2 d/8}}{e^{\ell\nu d}}. \quad \square$$

4.4 Completing the Lowerbound for GAP-TR$_{0,\tau}$ in the Local Model

In this section we complete the proof that in any ℓ-round, local, ϵ-differentially private protocols for the gap-threshold function, namely, GAP-TR$_{0,\tau}$, the curator errs with constant probability when $\ell \ll \sqrt{n}$ and τ is small. For proving this result, we defined a distribution $\mathcal{A}$ which chooses each bit in the input independently at random where it is one with probability α and zero with probability $1-\alpha$. Lemma 9, which follows from a standard Chernoff bound argument, states that when generating a vector $(X_1, \ldots, X_n)$ according to $\mathcal{A}$, the sum $\sum_{i=1}^{n} X_i$ is concentrated around its expected value, which is αn.

Lemma 9. $\Pr_{\mathcal{A}}\left[\sum_{i=1}^{n} X_i \leq (1-\gamma)\alpha n\right] < \exp\left(-\sqrt{dn}\gamma^2/(2\epsilon)\right)$ *for every* $0 \leq \gamma < 1$.

By Corollary 2 the distributions on the outputs when the inputs are taken from $\mathcal{A}$ or $\mathcal{B}$ are not far apart. By Lemma 9, with high probability the number of ones in the inputs distributed according to $\mathcal{A}$ is fairly big, while in $\mathcal{B}$ the number of ones is zero. These facts are used in Theorem 2 to prove the lowerbound.

Theorem 2. *In any ℓ-round, local, ϵ-differentially private protocol for computing* GAP-TR$_{0,\tau}$ *for $\tau = O(\sqrt{n}/(\epsilon\ell\sqrt{\log \ell}))$ the curator errs with constant probability,*

Proof. Fix any ℓ-round, local, ϵ-differentially private protocol, and let G be the algorithm of the curator that given the communication computes the output of the protocol. Let $\tau = 0.5\alpha n = \sqrt{dn}/\epsilon$. We denote $D \triangleq \{\mathbf{c} : G(\mathbf{c}) = 1\}$, that is, D contains all vectors of communication for which the curator answers 1. There are two cases. 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 2, the probability of D under the distribution $\mathcal{B}$ is big, and the curator has a big error when the inputs are distributed according to $\mathcal{B}$. Formally, there are two cases:

Case 1: $\Pr_{\mathcal{A}}[D] < 0.99$. We consider the event that the sum of the inputs is at least $\tau = 0.5\alpha n$ and the curator returns an answer 0, that is, the curator errs. We next prove 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 $\Pr_{\mathcal{A}}\left[\sum_{i=1}^{n} X_i < 0.5\alpha n\right] + \Pr_{\mathcal{A}}[D]$. By Lemma 9,

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

Thus, in this case, with probability ≈ 0.01 the curator errs.

Case 2: $\Pr_{\mathcal{A}}[D] \ge 0.99$. In this case, we consider the event that the input is the all-zero string and the curator answers 1, that is, the curator errs. We next prove using Corollary 2 that when the inputs are distributed according to $\mathcal{B}$ (that is, they are the all-zero string), the probability of this event is bounded away from 0, that is, taking $\nu = \theta(\ell \log \ell)$ and $d = 1/(\ell\nu) = \theta(1/(\ell^2 \log \ell)$,

$$\Pr_{\mathcal{B}}[D] \ge \frac{\Pr_{\mathcal{A}}[D] - \ell \exp\left(-(\nu - 8)^2 d/8\right)}{\exp\left(\ell \nu d\right)} > \frac{0.99 - 0.5}{\exp\left(1\right)} > 0.01.$$

Thus, in this case, with probability at least 0.01, the curator errs. As $d = \theta(1/(\ell^2 \log \ell))$, we get that $\tau = \sqrt{d}n/\epsilon = \theta(\sqrt{n}/(\epsilon\ell\sqrt{\log \ell}))$. $\square$

Acknowledgments. We thank Adam Smith 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 non-cryptographic fault-tolerant distributed computations. In *the 20th STOC*, pages 1–10, 1988.
2. A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the SuLQ framework. In *the 24th PODS*, pages 128–138, 2005.
3. D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols. In *the 20th STOC*, pages 11–19, 1988.
4. B. Chor and E. Kushilevitz. A communication-privacy tradeoff for modular addition. *Inform. Process. Lett.*, 45(4):205–210, 1993.
5. I. Damgård, M. Fitzi, E. Kiltz, J. B. Nielsen, and T. Toft. Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation. In *TCC 2006*, volume 3876 of *LNCS*, pages 285–304. 2006.
6. I. Dinur and K. Nissim. Revealing information while preserving privacy. In *the 22nd PODS*, pages 202–210, 2003.
7. C. Dwork. Differential privacy. In *the 33rd ICALP*, volume 4052 of *LNCS*, pages 1–12, 2006.

8. C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 486–503, 2006.
9. C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC 2006*, volume 3876 of *LNCS*, pages 265–284, 2006.
10. C. Dwork, F. McSherry, and K. Talwar. The price of privacy and the limits of LP decoding. In *39th STOC*, pages 85–94, 2007.
11. C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In *CRYPTO 2004*, volume 3152 of *LNCS*, pages 528–544, 2004.
12. A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *the 22nd PODS*, pages 211–222, 2003.
13. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *19th STOC*, pages 218–229, 1987.
14. S. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? Manuscipt, 2007.
15. F. McSherry and K. Talwar. Mechanism design via differential privacy. In *the 48th FOCS*, pages 94–103, 2007.
16. K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *the 39th STOC*, pages 75–84, 2007.
17. S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.
18. A. C. Yao. Protocols for secure computations. In *the 23th FOCS*, pages 160–164, 1982.

A A Composition Lemma

Assume an interactive protocol where a (deterministic) curator C makes adaptive queries to a party p holding a private input $x \in \{0,1\}$. I.e., for $0 \le i \le \ell$, in round $2i$ the curator sends p a message $A_i = C(i, \mathcal{V}_1, \ldots, \mathcal{V}_{i-1})$ computed over the transcript of messages $\mathcal{V}_1, \ldots, \mathcal{V}_{i-1}$ already received from p, and specifying a randomized algorithm A_i; in round $2i+1$ party p computes $\mathcal{V}_i = A_i(x)$ (using fresh random coins for each A_i) and sends $\mathcal{V}_i$ to C.

Definition 4. *A possible outcome $\mathcal{V}$ is ϵ-good for algorithm A if* $\Pr[A(1) = \mathcal{V}] \le e^{\epsilon} \Pr[A(0) = \mathcal{V}]$*, where the probabilities are taken over the randomness of algorithm A. An algorithm A is (ϵ, δ)-good if* $\Pr[A(1)$ *is ϵ-good for A*$] \ge 1 - \delta$*, where the probability is taken over the randomness of A.*

Assume that the range of C only includes (ϵ, δ)-good algorithms. 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 $(A_1, \mathcal{V}_1, A_2, \mathcal{V}_2, \ldots, A_\ell, \mathcal{V}_\ell)$ sampled according to the protocol above.

Lemma 10. *$\hat{A}$ is $(\ell\epsilon, 1 - (1-\delta)^\ell)$-good.*

Proof. Note first, that with probability at least $(1-\delta)^\ell$, the result of $\hat{A}(1)$ is a transcript $\hat{\mathcal{V}} = (A_1, \mathcal{V}_1, A_2, \mathcal{V}_2, \ldots, A_\ell, \mathcal{V}_\ell)$ such that $\mathcal{V}_i$ is ϵ-good for A_i for all $i \le \ell$. It suffices, hence, to prove that when that happens the transcript $\hat{\mathcal{V}}$ is $\ell\epsilon$-good for $\hat{A}$, and indeed: $\Pr[\hat{A}(1) = (A_1, \mathcal{V}_1, A_2, \mathcal{V}_2, \ldots, A_\ell, \mathcal{V}_\ell)] = \prod_{i=1}^{\ell} \Pr[A_i(1) = \mathcal{V}_i] \le \prod_{i=1}^{\ell} e^{\epsilon} \cdot \Pr[A_i(0) = \mathcal{V}_i] = e^{\ell\epsilon} \cdot \Pr[\hat{A}(0) = (A_1, \mathcal{V}_1, A_2, \mathcal{V}_2, \ldots, A_\ell, \mathcal{V}_\ell)]$. □