On Communication Protocols that Compute Almost Privately[☆]

Marco Comi[a], Bhaskar DasGupta[a,*], Michael Schapira[b], Venkatakumar Srinivasan[a]

[a] *Department of Computer Science, University of Illinois at Chicago, IL 60607*
[b] *School of Computer Science and Engineering, The Hebrew University of Jerusalem, 91904 Jerusalem, Israel*

Abstract

We further investigate and generalize the approximate privacy model recently introduced by Feigenbaum *et al.* [8]. We explore the privacy properties of a natural class of communication protocols that we refer to as "*dissection protocols*". Informally, in a dissection protocol the communicating parties are restricted to answering questions of the form "*Is your input between the values* α *and* β (under a pre-defined order over the possible inputs)?". We prove that for a large class of functions, called *tiling functions*, there *always* exists a dissection protocol that provides a *constant average-case privacy approximation ratio* for uniform or "almost uniform" probability distributions over inputs. To establish this result we present an interesting connection between the approximate privacy framework and basic concepts in computational geometry. We show that such a good privacy approximation ratio for tiling functions does *not*, in general, exist in the *worst case*. We also discuss extensions of the basic setup to more than two parties and to non-tiling functions, and provide calculations of privacy approximation ratios for two functions of interest.

Keywords: Approximate privacy, multi-party communication, tiling,

[☆] A preliminary version of this paper appeared in the 4th Symposium on Algorithmic Game Theory, G. Persiano (Ed.), LNCS 6982, Springer-Verlag, 44-56, 2011.

[*] Corresponding author. Phone: 312-355-1319, Fax: 312-413-0024.

Email addresses: ingmarco85@gmail.com (Marco Comi), bdasgup@uic.edu (Bhaskar DasGupta), schapiram@huji.ac.il (Michael Schapira), vsrini7@uic.edu (Venkatakumar Srinivasan)

URL: http://www.cs.uic.edu/~dasgupta (Bhaskar DasGupta)

binary space partition
2010 MSC: 68Q01, 68M14

1. Introduction

Consider the following interaction between two parties, Alice and Bob. Each of the two parties, Alice and Bob, holds a *private* input, x_{bob} and y_{alice} respectively, not known to the other party. The two parties aim to compute a function f of the two private inputs. Alice and Bob alternately query each other to make available a *small* amount of information about their private inputs, *e.g.*, an answer to a range query on their private inputs or a few bits of their private inputs. This process ends when each of them has seen enough information to be able to compute the value of $f(x_{\mathrm{bob}}, y_{\mathrm{alice}})$. The central question that is the focus of this paper is:

> *Can we design a communication protocol whose execution reveals, to both Alice and Bob, as well as to any eavesdropper, as little information as possible about the other's private input beyond what is necessary to compute the function value*?

Note that there are two conflicting constraints: Alice and Bob need to communicate sufficient information for computing the function value, but would prefer not to communicate too much information about their private inputs. This setting can be generalized in an obvious manner to $d > 1$ parties $\mathsf{party}_1, \mathsf{party}_2, \ldots, \mathsf{party}_d$ computing a d-ary f by querying the parties in round-robin order, allowing each party to broadcast information about its private input (via a public communication channel).

Privacy preserving computational models such as the one described above have become an important research area due to the increasingly widespread usage of sensitive data in networked environments, as evidenced by distributed computing applications, game-theoretic settings (*e.g.*, auctions) and more. Over the years computer scientists have explored many *quantifications* of privacy in computation. Much of this research focused on designing *perfectly* privacy-preserving protocols, *i.e.*, protocols whose execution reveals *no* information about the parties' private inputs beyond that implied by the outcome of the computation. Unfortunately, perfect privacy is often either *impossible*, or *infeasibly costly* to achieve (*e.g.*, requiring impractically extensive communication steps). To overcome this, researchers have also investigated various notions of *approximate privacy* [7, 8].

In this paper, we adopt the approximate privacy framework of [8] that quantifies approximate privacy via the *privacy approximation ratios* (Pars) of protocols for computing a deterministic function of two private inputs. *Informally*, Par captures the objective that an observer of the transcript of the entire protocol will not be able to distinguish the real inputs of the two communicating parties from *as large a set as possible* of other inputs. To capture this intuition, [8] makes use of the machinery of communication-complexity theory to provide a geometric and combinatorial interpretation of protocols. [8] formulates both the worst-case and the average-case version of Pars and studies the tradeoff between privacy preservation and communication complexity for several functions of interest.

1.1. Motivations from Mechanism Designs

An original motivation of this line of research, as explained in details in [8], comes from privacy concerns in auction theory in Economics. A traditionally desired goal of designing auction mechanisms is to ensure that it is *incentive compatible*, *i.e.*, bidders fare best by letting their truthful bids known. However, more recently, another *complementary* goal that has gained significant attention, specially in the context of online auctions, is to preserve privacy of the bidders, *i.e.*, bidders reveal as little information as necessary to auctioneers for optimal outcomes. To give an example, consider a 2^{nd}-price Vickrey auction of an item via a straightforward protocol in which the price of the item is incrementally increased until the winner is determined. However, the protocol reveals more information than what is *absolutely necessary*, namely the information about the identity of the winner (with revealing his/her bid) together with the bid of the second-highest bidder, and revealing such additional information could put the winner at a disadvantage in the bidding process of a similar item in the future since the auctioneer could set a lower reserve price. In this paper, we consider a generalized version of the setting that captures applications of the above type as well as other applications in multi-party computation.

2. Summary of Our Contributions

Any investigation of approximate privacy for multi-party computation starts by defining how we quantify approximate privacy. In this paper, we use the combinatorial framework of [8] for quantification of approximate privacy for two parties via PARs and present its natural extension to three or more parties. Often, parties' inputs have a natural ordering, *e.g.*, the private input of a party belongs to some range of integers $\{L, L+1, \ldots, M\}$ (as is the case when computing, say, the maximum or minimum of two inputs). When designing protocols for such environments, a natural restriction is to only allow the protocol to ask each party questions of the form "*Is your input between the values α and β (under this natural order over possible inputs)?*". We refer to this type of protocols as *dissection protocols* and study the privacy properties of this natural class of protocols. We note that the bisection and c-bisection protocols for the millionaires problem and other problems in [8], as well as the bisection auction in [9, 10], all fall within this category of protocols. Our findings are summarized below.

Average- and worst-case PARs for tiling functions for two party computation. We first consider a broad class of functions, referred to as the *tiling functions* in the sequel, that encompasses several well-studied functions (*e.g.*, Vickrey's second-price auctions). Informally, a two-variable tiling function is a function whose output space can be viewed as a collection of disjoint combinatorial rectangles in the two-dimensional plane, where the function has the same value within each rectangle. A first natural question for investigation is to classify those tiling functions for which there exists a perfectly privacy-preserving dissection protocol. We observe that for every Boolean tiling functions (*i.e.*, tiling functions which output binary values) *this is indeed the case.* In contrast, for tiling functions with a range of just three values, perfectly privacy-preserving computation is no longer necessarily possible (even when not restricted to dissection protocols).

We next turn our attention to PARs. We prove that for *every* tiling function there exists a dissection protocol that achieves a constant PAR in the average case (that is, when the parties' private values are drawn from an uniform or *almost* uniform probability distribution). To establish this result, we make use of results on the binary space partitioning problems studied in the computational geometry literature. We complement this positive result for dissection protocols with the following negative result: *there exist tiling functions for which no dissection protocol can achieve a constant* PAR *in the worst-case.*

Extensions to non-tiling functions and three-party communication. We discuss two extensions of the above results. We explain how our constant average-case PAR result for tiling functions can be extended to a family of "almost" tiling functions. In addition, we consider the case of *more than two* parties. We show that in this setting it is *no longer true* that for every tiling function there exists a dissection protocol that achieves a constant PAR in the average case. Namely, we exhibit a three-dimensional tiling function for which *every* dissection protocol exhibits *exponential* average- and worst-case PARs, *even when an unlimited number of communication steps is allowed.*

PARs for the set covering and equality functions. [8] presents bounds on the average-case and the worst-case PARs of the bisection protocol — a special case of dissection protocols — for several functions (Yao's millionaires' problem, Vickrey's second-price auction, and others). We analyze the PARs of the bisection protocol for two well-studied Boolean functions: the set-covering and equality functions; the equality function provides a useful testbed for evaluating privacy preserving protocols [3] [11, Example 1.21] and set-covering type of functions are useful for studying the differences between deterministic and non-deterministic communication complexities [11, Section 10.4]. We show that, for both functions, the bisection protocol *fails to achieve* good PARs in both the average-case and the worst-case.

3. Summary of Prior Related Works

3.1. Privacy-preserving Computation

Privacy-preserving computation has been the subject of extensive research and has been approached from information-theoretic [3], cryptographic [5], statistical [12], communication complexity [13, 17], statistical database query [7] and other perspectives [11]. Among these, most relevant to our work is the approximate privacy framework of Feigenbaum *et al.* [8] that presents a metric for quantifying privacy preservation building on the work of Chor and Kushilevitz [6] on characterizing perfectly privately computable computation and on the work of Kushilevitz [13] on the communication complexity of perfectly private computation. The bisection, c-bisection and bounded bisection protocols of [8] fall within our category of dissection protocol since we allow the input space of each party to be divided into two subsets of arbitrary size. There are also some other formulations of perfectly and approximately privacy-preserving computation in the literature, but they are inapplicable in our context. For example, the differential privacy model (see [7]) approaches privacy in a different context via adding noise to the result of a database query in such a way as to preserve the privacy of the individual database records but still have the result convey nontrivial information,

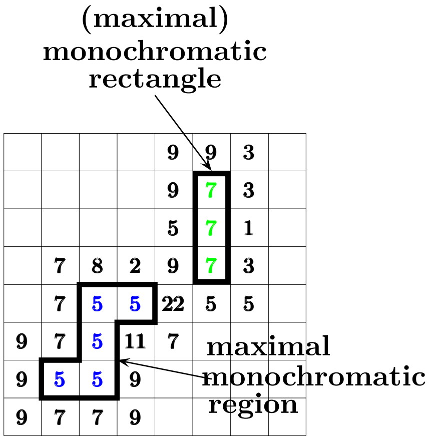
*3.2. Binary space partition (*Bsp*)*

Bsps present a way to implement a *geometric divide-and-conquer* strategy and is an extremely popular approach in numerous applications such as hidden surface removal, ray-tracing, visibility problems, solid geometry, motion planning and spatial databases (*e.g.*, see [16]). However, to the best of our knowledge, a connection between Bsps bounds such as in [14, 15, 2, 4] and approximate privacy has not been explored before.

4. The Model and Basic Definitions

4.1. Two-party Approximate Privacy Model of [8]

We have two parties party_1 and party_2, each a binary string, x_1 and x_2 respectively, which represents a private value in some set $\mathcal{U}^{\text{in}}$. The common goal of the two parties is to compute the value $f(x_1, x_2)$ of a given public-knowledge two-variable function f. Before a communication protocol P starts, each party_i initializes its "*set of maintained inputs*" $\mathcal{U}_i^{\text{in}}$ to $\mathcal{U}^{\text{in}}$. In one step of communication, one party transmits a bit indicating in which of two parts of its input space its private input lies. The other party then updates its set of maintained inputs accordingly. The very last information transmitted in the protocol P contains the value of $f(x_1, x_2)$. The final transcript of the protocol (*i.e.*, the entire information exchanged) is denoted by $s(x_1, x_2)$.



				9	9	3	
				9	7	3	
				5	7	1	
	7	8	2	9	7	3	
	7	5	5	22	5	5	
9	7	5	11	7			
9	5	5	9				
9	7	7	9				

Figure 1: *An illustration of some communication-complexity definitions.*

Denoting the domain of outputs by $\mathcal{U}^{\text{out}}$, any function $f : \mathcal{U}^{\text{in}} \times \mathcal{U}^{\text{in}} \mapsto \mathcal{U}^{\text{out}}$ can be visualized as $\left|\mathcal{U}^{\text{in}}\right| \times \left|\mathcal{U}^{\text{in}}\right|$ matrix with entries from $\mathcal{U}^{\text{out}}$ in which the first dimension represents the possible values of party_1, ordered by some permutation Π_1, while the second dimension represents the possible values of party_2, ordered by some permutation Π_2, and each entry contains the value of f associated with a particular set of inputs from the two parties. This matrix will be denoted by $A_{\Pi_1,\Pi_2}(f)$, or sometimes simply by A.

We now present the following definitions from [11, 8]; see Fig. 1 for a geometric illustration.

Definition 1 (Regions, partitions). *A region of A is any subset of entries in A. A partition of A is a collection of disjoint regions in A whose union equals to A*

Definition 2 (Rectangles, tilings, refinements). *A rectangle in A is a submatrix of A. A tiling of A is a partition of A into rectangles. A tiling T_1 of A is a refinement of another tiling T_2 of A if every rectangle in T_1 is contained in some rectangle in T_2.*

Definition 3 (Monochromatic, maximal monochromatic and ideal monochromatic partitions). *A region R of A is monochromatic if all entries in R are of the same value. A monochromatic partition of A is a partition all of whose regions are monochromatic. A monochromatic region of A is a maximal monochromatic region if no monochromatic region in A properly contains it. The ideal monochromatic partition of A is made up of the maximal monochromatic regions.*

Definition 4 (Perfect privacy). *Protocol P achieves perfect privacy if, for every two sets of inputs (x_1, x_2) and (x_1', x_2') such that $f(x_1, x_2) = f(x_1', x_2')$, it holds that $s(x_1,\ x_2) = s(x'_1,\ x'_2)$. Equivalently, a protocol P for f achieves perfectly privacy if the monochromatic tiling induced by P is the ideal monochromatic partition of $A(f)$.*

Definition 5 (Worst case and average case PAR of a protocol P). *Let $R^P(x_1, x_2)$ be the monochromatic rectangle containing the cell $A(x_1, x_2)$ induced by P, $R^I(x_1, x_2)$ be the monochromatic region containing the cell $A(x_1, y_1)$ in the ideal monochromatic partition of A, and $\mathcal{D}$ be a probability distribution over the space of inputs. Then P has a worst-case* PAR *of α_{worst} and an average case* PAR *of $\alpha_{\mathcal{D}}$ under distribution $\mathcal{D}$ provided*[1]

$$\alpha_{\text{worst}} = \max_{(x_1,x_2)\in\mathcal{U}^{\text{in}}\times\mathcal{U}^{\text{in}}} \frac{|\, R^I(x_1,x_2)\,|}{|\, R^P(x_1,x_2)|} \ \ \textit{and}\ \ \alpha_{\mathcal{D}} = \sum_{(x_1,x_2)\in\mathcal{U}^{\text{in}}\times\mathcal{U}^{\text{in}}} \Pr_{\mathcal{D}}[x_1 \,\&\, x_2]\, \frac{|R^I(x_1,x_2)|}{|R^P(x_1,x_2)|}$$

[1] The notation $\Pr_{\mathcal{D}}[\mathcal{E}]$ denotes the probability of an event $\mathcal{E}$ under distribution $\mathcal{D}$.

Definition 6 (PAR for a function). *The worst-case (average-case)* PAR *for a function f is the minimum, over all protocols P for f, of the worst-case (average-case)* PAR *of P.*

Extension to Multi-party Computation. In the multi-party setup, we have $d > 2$ parties $\mathsf{party}_1, \mathsf{party}_2, \ldots, \mathsf{party}_d$ computing a d-ary function $f : (\mathcal{U}^{\text{in}})^d \mapsto \mathcal{U}^{\text{out}}$. Now, f can be visualized as $|\mathcal{U}^{\text{in}}| \times \cdots \times |\mathcal{U}^{\text{in}}|$ matrix $A_{\Pi_1,\ldots,\Pi_d}(f)$ (or, sometimes simply by A) with entries from $\mathcal{U}^{\text{out}}$ in which the i^{th} dimension represents the possible values of party_i ordered by some permutation Π_i, and each entry of A contains the value of f associated with a particular set of inputs from the d parties. Then, all the previous definitions can be naturally adjusted in the obvious manner, *i.e.*, the input space as a d-dimensional space, each party maintains the input partitions of all other $d-1$ parties, the transcript of the protocol s is a d-ary function, and rectangles are replaced by d-dimensional hyper-rectangles (Cartesian product of d intervals).

4.2. Dissection Protocols and Tiling Functions for Two-party Computation

Often in a communication complexity settings the input of each party has a natural ordering, *e.g.*, the set of input of a party from $\{0,1\}^k$ can represent the numbers $0, 1, 2, \ldots, 2^k - 1$ (as is the case when computing the maximum/minimum of two inputs, in the millionaires problem, in second-price auctions, and more). When designing protocols for such environments, a natural restriction is to only the allow protocols such that each party asks questions of the form "*Is your input between a and b (in this natural order over possible inputs)?*", where $a, b \in \{0,1\}^k$. Notice that after applying an appropriate permutation to the inputs, such a protocol divides the input space into two (not necessarily equal) halves. Below, we formalize these types of protocols as "*dissection protocols*".

Definition 7 (contiguous subset of inputs). *Given a permutation Π of $\{0,1\}^k$, let $\prec_\Pi$ denote the total order over $\{0,1\}^k$ that Π induces,* i.e., *$\forall\, a, b \in \{0,1\}^k$, $a \prec_\Pi b$ provided b comes after a in Π. Then, $I \subseteq \{0,1\}^k$* contiguous *with respect to Π if $\forall\, a, b \in I$, $\forall\, c \in \{0,1\}^k$: $a \prec_\Pi c \prec_\Pi b \Longrightarrow c \in I$.*

Definition 8 (dissection protocol). *Given a function $f : \{0,1\}^k \times \{0,1\}^k \mapsto \{0,1\}^t$ and permutations Π_1, Π_2 of $\{0,1\}^k$, a protocol for f is a* dissection protocol *with respect to (Π_1, Π_2) if, at each communication step, the maintained subset of inputs of each party_i is contiguous with respect to Π_i.*

Observe that *every* protocol P can be regarded as a dissection protocol with respect to *some* permutations over inputs by simply constructing the permutation so that it is consistent with the way P updates the maintained sets of inputs. However, *not* every protocol is a dissection protocol with respect to *specific* permutations. Consider, for example, the case that both Π_1 and Π_2 are the permutation over $\{0,1\}^k$ that orders the elements from lowest to highest binary values. Observe that a protocol that is a dissection protocol with respect to these permutations *cannot* ask questions of the form "Is your input odd or even?", for these questions partition the space of inputs into *non-contiguous* subsets with respect to (Π_1, Π_2).

A special case of interest of the dissection protocol is the "bisection type" protocols that have been investigated in the literature in many contexts [8, 10].

Definition 9 (bisection, c-bisection and bounded-bisection protocols)**.** *For a constant $c \in \left[\frac{1}{2}, 1\right)$, a dissection protocol with respect to the permutations (Π_1, Π_2) is called a c-bisection protocol provided at each communication step each party_i partitions its input space of size z into two halves of size $c\,z$ and $(1-c)\,z$. A bisection protocol is simply a $\frac{1}{2}$-bisection protocol. For an integer valued function $g(k)$ such that $0 \leq g(k) \leq k$, bounded-bisection$_{g(k)}$ is the protocol that runs a bisection protocol with $g(k)$ bisection operations followed by a protocol (if necessary) in which each party_i repeatedly partitions its input space into two halves one of which is of size exactly one.*

We next introduce the concept of *tiling* functions.

Definition 10 (tiling and non-tiling functions)**.** *A function $f : \{0,1\}^k \times \{0,1\}^k \mapsto \{0,1\}^t$ is called a tiling function with respect to two permutations (Π_1, Π_2) of $\{0,1\}^k$ if the monochromatic regions in $A_{\Pi_1,\Pi_2}(f)$ form a tiling, and the number of monochromatic regions in this tiling is denote by $\mathsf{r}_f\,(\Pi_1, \Pi_2)$. Conversely, f is a non-tiling function if f is not a tiling function with respect to every pair of permutations (Π_1, Π_2) of $\{0,1\}^k$.*

For example, $f(x_1, \ldots, x_k, y_1, \ldots, y_k) \equiv \sum_{i=1}^{k} (x_i + y_i) \pmod 2$ is a tiling function with respect to (Π_1, Π_2) with $\mathsf{r}_f\,(\Pi_2, \Pi_2) = 4$, where each Π_i orders its inputs $(z_1, \ldots, z_k)$ in increasing order of $\sum_{i=1}^{k} z_i \pmod 2$. Note that a function f that is tiling function with respect to permutations (Π_1, Π_2) may not be a tiling function with respect to a different set of permutations (Π'_1, Π'_2); see Fig. 3. Also, a function f can be a tiling function with respect

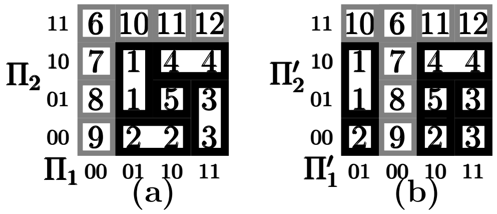


Figure 2: *A tiling function with respect to different permutation pairs* (Π_1, Π_2) *and* (Π'_1, Π'_2) *inducing different numbers of monochromatic rectangles.*

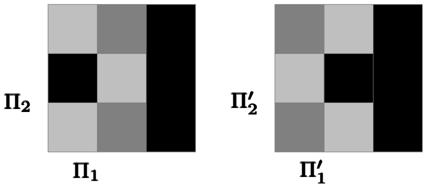


Figure 3: *Tilability depends on* Π_1 *and* Π_2*.*

to two distinct permutation pairs (Π_1, Π_2) and (Π'_1, Π'_2), and the number of monochromatic regions in the two cases differ; see Fig. 2. Thus, indeed we need Π_1 and Π_2 in the definition of tiling functions and r_f.

Extensions to Multi-party Computation. For the multi-party computation model involving $d > 2$ parties, the d-ary tiling function f has a permutation Π_i of $\{0,1\}^k$ for each i^{th} argument of f (or, equivalently for each party_i). A dissection protocol is generalized to a "round robin" dissection protocol in the following manner. In one "mega" round of communications, parties communicate in a fixed order, say $\mathsf{party}_1, \mathsf{party}_2, \ldots, \mathsf{party}_d$, and the mega round is repeated if necessary. Any communication by any party is made available to *all* the other parties. Thus, each communication of the dissection protocol partitions a d-dimensional space by an appropriate *set* of $(d-1)$-dimensional hyperplanes, where the missing dimension in the hyperplane correspond to the index of the party communicating.

5. Two-party Dissection Protocol for Tiling Functions

5.1. Boolean Tiling Functions

Lemma 1. *Any Boolean tiling function* $f\colon \{0,1\}^k \times \{0,1\}^k \mapsto \{0,1\}$ *with respect to some two permutations* (Π_1, Π_2) *can be computed in a perfectly privacy-preserving manner by a dissection protocol with respect to the same permutations* (Π_1, Π_2)*.*

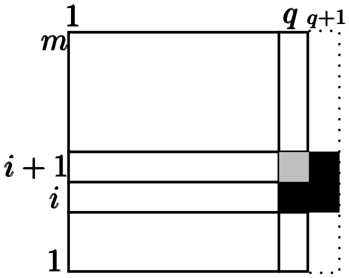


Figure 4: *This configuration cannot happen in Case* **2**.

Proof. For any $m \times n$ Boolean matrix A with rows and columns indexed by $1, 2, \ldots, m$ and $1, 2, \ldots, n$, respectively, let the notation $A[i_1, i_2, j_1, j_2]$ denote the submatrix of A consisting of rows $i_1, i_1 + 1, \ldots, i_2$ and columns $j_1, j_1 + 1, \ldots, j_2$. Assume $m, n \geq 2$ and suppose that the zeroes and ones in the matrix A form a tiling. We claim that there must exist an index $i \in \{1, 2, \ldots, m-1\}$ such that the partition of A into the two sub-matrices $A[1, i, 1, n]$ and $A[i+1, m, 1, n]$ does not split any tile, or that there must exist an index $j \in \{1, 2, \ldots, n-1\}$ such that the partition of A into the two sub-matrices $A[1, m, 1, j]$ and $A[1, m, j+1, n]$ does not split any tile. This claim, applied recursively on each submatrix of A, will prove Lemma 1.

We prove our claim by induction on n. The basis case of $n = 2$ follows trivially. Suppose that our claim is true for all $n \in \{2, \ldots, q\}$ and consider the case of $n = q + 1$.

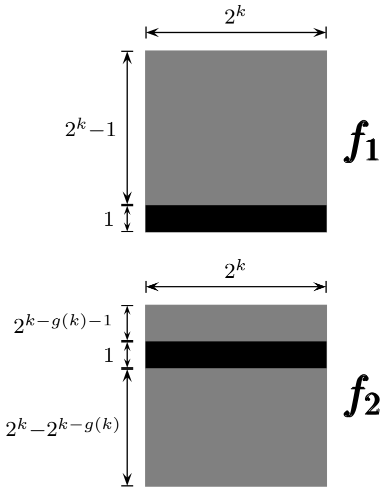
Case 1: there exists an index $j \in \{1, 2, \ldots, q-1\}$ such that the partition of $A[1, m, 1, q]$ into the two sub-matrices $A[1, m, 1, j]$ and $A[1, m, j+1, q]$ does not split any tile. Then, the same index j works for $A[1, m, 1, q+1]$ also.

Case 2: there is no such index j as in Case **1** above, but there exists an index $i \in \{1, 2, \ldots, m-1\}$ such that the partition of $A[1, m, 1, q]$ into the two submatrices $A[1, i, 1, q]$ and $A[i+1, m, 1, q]$ does not split any tile. Suppose that the index i does split a tile in the partition $A[1, i, 1, q+1]$ and $A[i+1, m, 1, q+1]$ of $A[1, m, 1, q+1]$. Then, we must have the situation as shown in Fig. 4, which shows that the zeroes and ones of $A[1, m, 1, q+1]$ do not form a tiling. ❑

Remark 1. *As Fig. 4 shows, the claim of Lemma 1 is false if f outputs three values.*

5.2. Average and Worst Case Par *for Non-Boolean Tiling Functions*

Let $f : \{0,1\}^k \times \{0,1\}^k \mapsto \{0,1\}^t$ be a given tiling function with respect to permutations (Π_1, Π_2). Neither the c-bisection nor the bounded-bisection protocol performs well in terms of average Par on arbitrary tiling functions; see Fig. 5 for an illustration. In this section, we show that *any* tiling function



protocol	number of steps of communication	average case PAR for distribution D_u
c-bisection on f_1 $(\, c \in [1/2, 1) \,)$	$k/\log_2 \frac{1}{c}$	k
bounded-bisection$_{g(k)}$ on f_2 $(\, 1 \le g(k) \le k \,)$	$g(k) + 2^{k-g(k)} - 1$	$g(k) + 2^{k-g(k)-1} - 1$

Figure 5: *Functions f_1 and f_2 with $\mathsf{r}_{f_1}(\Pi_1, \Pi_2) = \mathsf{r}_{f_2}(\Pi_1, \Pi_2) = 2$. The bisection-type protocols fail to achieve a good average-case* PAR *on them.*

f admits a dissection protocol that has a *small constant* average case PAR. Moreover, we show that this result *cannot* be extended to the case of worst-case PARS.

5.2.1. Constant Average-case PAR *for Non-Boolean Functions*

Let D_u denote the uniform distribution over all input pairs. We define the notion of a c-approximate uniform distribution $\mathsf{D}_u^{\sim c}$; note that $\mathsf{D}_u^{\sim 0} \equiv \mathsf{D}_u$.

Definition 11 (c-approximate uniform distribution)**.** *A c-approximate uniform distribution $\mathsf{D}_u^{\sim c}$ is a distribution in which the probabilities of the input pairs are close to that for the uniform distribution as a linear function of c, namely*

$$\max_{(\mathbf{x},\mathbf{y}),\,(\mathbf{x}',\mathbf{y}')\in\{0,1\}^k\times\{0,1\}^k} \left| \Pr_{\mathsf{D}_u^{\sim c}}[\mathbf{x}\,\&\,\mathbf{y}] - \Pr_{\mathsf{D}_u^{\sim c}}[\mathbf{x}'\,\&\,\mathbf{y}'] \right| \le c\,2^{-2k}$$

Theorem 1.
(a) *A tiling function f with respect to permutations (Π_1, Π_2) admits a dissection protocol P with respect to the same permutations (Π_1, Π_2) using at most $4\,\mathsf{r}_f(\Pi_1, \Pi_2)$ communication steps such that $\alpha_{\mathsf{D}_{\tilde{u}}^{c}} \leq 4 + 4\,c$.*

(b) *For all $0 \leq c < 1$, there exists a tiling function $f\colon \{0,1\}^2 \times \{0,1\}^2 \mapsto \{0,1\}^4$ such that, for any two permutations (Π_1, Π_2) of $\{0,1\}^2$, every dissection protocol with respect to (Π_1, Π_2) using any number of communication steps has $\alpha_{\mathsf{D}_{\tilde{u}}^{c}} \geq \frac{9+c}{8}$.*

Proof. Let $\mathcal{S} = \{S_1, S_2, \ldots, S_{\mathsf{r}_f}\}$ be the set of $\mathsf{r}_f = \mathsf{r}_f(\Pi_1, \Pi_2)$ ideal monochromatic rectangles in the tiling of f induced by the permutations (Π_1, Π_2) and consider a protocol P that is a dissection protocol with respect to (Π_1, Π_2). Suppose that the ideal monochromatic rectangle $S_i \in \mathcal{S}$ has y_i elements, and P partitions this rectangle into t_i rectangles $S_{i,1}, \ldots, S_{i,t_i}$ having $z_{i,1}, \ldots, z_{i,t_i}$ elements, respectively. Then, using the definition of α_{D_u} it follows that

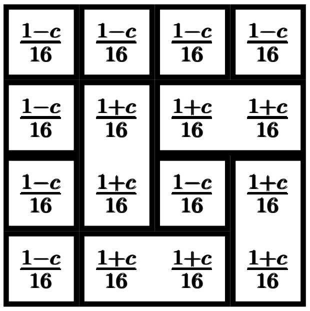
$$\alpha_{\mathsf{D}_u} = \sum_{(x_1,x_2)\in\mathcal{U}\times\mathcal{U}} \Pr_{\mathsf{D}_u}[x_1 \,\&\, x_2]\, \frac{|R^I(x_1,x_2)|}{|R^P(x_1,x_2)|}$$

$$= \sum_{i=1}^{\mathsf{r}_f} \sum_{j=1}^{t_i} \sum_{(x_1,x_2)\in S_{i,j}} \Pr_{\mathsf{D}_u}[x_1 \,\&\, x_2]\, \frac{y_i}{z_{i,j}} = \sum_{i=1}^{\mathsf{r}_f} \sum_{j=1}^{t_i} \frac{y_i}{2^{2k}} = \sum_{i=1}^{\mathsf{r}_f} \frac{t_i\, y_i}{2^{2k}}$$

Similarly, it follows that

$$\alpha_{\mathsf{D}_{\tilde{u}}^{c}} \leq \sum_{i=1}^{\mathsf{r}_f} \sum_{j=1}^{t_i} \sum_{(x_1,x_2)\in S_{i,j}} \frac{1+c}{2^{2k}} \times \frac{y_i}{z_{i,j}} = \sum_{i=1}^{\mathsf{r}_f} \sum_{j=1}^{t_i} \frac{(1+c)\, y_i}{2^{2k}} = \sum_{i=1}^{\mathsf{r}_f} \frac{(1+c)\, t_i\, y_i}{2^{2k}}$$

A binary space partition (Bsp) for a collection of *disjoint* rectangles in the two-dimensional plane is defined as follows. The plane is divided into two parts by cutting rectangles with a line if necessary. The two resulting parts of the plane are divided recursively in a similar manner; the process continues until at most one fragment of the original rectangles remains in any part of the plane. This division process can be naturally represented as a binary tree (Bsp-tree) where a node represents a part of the plane and stores the cut that splits the plane into two parts that its two children represent and each leaf of the Bsp-tree represents the final partitioning of the plane by storing at most one fragment of an input rectangle; see Fig. 7 for an illustration. The *size* of a Bsp is the *number of leaves* in the Bsp-tree. The following result is known.



$\frac{1-c}{16}$	$\frac{1-c}{16}$	$\frac{1-c}{16}$	$\frac{1-c}{16}$
$\frac{1-c}{16}$	$\frac{1+c}{16}$	$\frac{1+c}{16}$	$\frac{1+c}{16}$
$\frac{1-c}{16}$	$\frac{1+c}{16}$	$\frac{1-c}{16}$	$\frac{1+c}{16}$
$\frac{1-c}{16}$	$\frac{1+c}{16}$	$\frac{1+c}{16}$	$\frac{1+c}{16}$

Figure 6: *Example for* $\alpha_{\mathrm{D}_{\tilde{u}}^{c}} \geq \frac{9+c}{8}$. *The tiles are shown by thick black lines. The numbers shown at each cell is the associated probability of that input.*

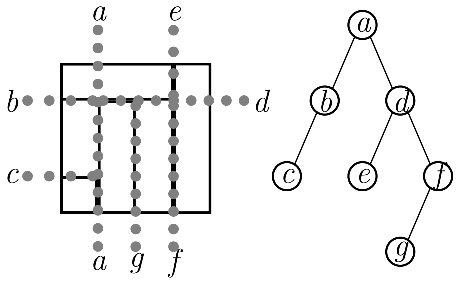


Figure 7: Bsp and Bsp-tree.

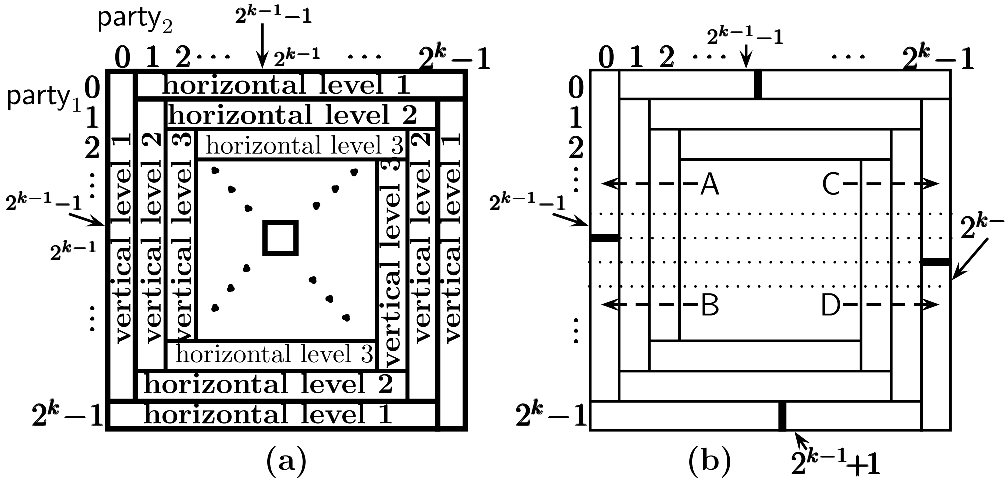


Figure 8: *Illustrations of the arguments in the proof of Theorem 2. The dotted lines in* **(b)** *are shown for visual clarities only.*

Fact 1. [4] *Assume that we have a set $\mathcal{S}$ of disjoint axis-parallel rectangles in the plane. Then, there is a* Bsp *of $\mathcal{S}$ such that every rectangle in $\mathcal{S}$ is partitioned into at most* 4 *rectangles.*[2]

(a) Consider the dissection protocol corresponding to the Bsp in Fact 1.

[2] The stronger bounds by Berman, DasGupta and Muthukrishnan [2] apply to *average* number of fragments only.

Then, using $\max_i\{t_i\} \leq 4$ we get $\alpha_{\mathrm{D}_{\tilde{u}}^{c}} \leq \sum_{i=1}^{\mathsf{r}_f} \frac{4\,(1+c)\,y_i}{2^{2k}} = 4\,(1+c)$. Also, the number of communication steps in this protocol is the height of the Bsp-tree, which is at most $4\mathsf{r}_f$.

(b) Consider the function f whose ideal monochromatic rectangles are shown in Fig. 6. Any correct protocol for computing f *must* partition at least one rectangle of two elements, giving

$$\alpha_{\mathrm{D}_{\tilde{u}}^{c}} \geq 4 \times \left(\frac{1+c}{16}\right) + \frac{7-c}{8} = \frac{9+c}{8}$$

❑

5.2.2. Large Worst-case Par *for Non-Boolean Functions*

Can one extend the results of the last section to show that every tiling function admits a dissection protocol that achieves a good Par *even in the worst case*? We answer this question in the negative by presenting a tiling function for which *every* dissection protocol has *large* worst-case Par.

Theorem 2. *Let $k > 0$ be an even integer. Then, there exists a tiling function $f : \{0,1\}^k \times \{0,1\}^k \mapsto \{0,1\}^3$ with respect to some two permutations (Π_1, Π_2) such that, for any two permutations Π_1' and Π_2' of $\{0,1\}^k$, every dissection protocol for f with respect to (Π_1', Π_2') has $\alpha_{\text{worst}} > 2^{k/2} - 1$.*

Proof. Recall the example in Fig. 6 that essentially showed that there exists functions that cannot be computed in a perfectly private manner. Our construction of the function f is based on the tiling shown in Fig. 6. We consider the specific permutations Π_1, Π_2 over $\{0,1\}^k$ that order the elements in $\{0,1\}^k$ by binary value (from 0 to $2^k - 1$). We now use the construction in Fig. 6 "recursively" to create a tiling of the input space. We first embed $\frac{2^k-2}{2} = 2^{k-1} - 1$ instances of the construction in Figure 6 recursively within one another, as shown in Fig. 8(a), leaving a 1×1 square at the center. The vertical level i and the horizontal level i rectangles have dimension $1 \times \big(2^k - (2i-1)\big)$ and $\big(2^k - (2i-1)\big) \times 1$, respectively, for $i = 1, 2, \ldots, 2^{k-1} - 1$. We then partition each of the level 1 rectangle in Fig. 8(a) into two "nearly" equal-sized rectangles as shown in Fig. 8(b). Consider the function f such that the monochromatic rectangles of $A_f(\Pi_1, \Pi_2)$ are the tilings in Fig. 8(b) (f outputs a different outcome for each (minimal) rectangle in the figure). Clearly, f is a tiling function with respect to

(Π_1, Π_2) and, moreover, since every rectangle shares a side with no more than 8 rectangles, at most 8 output values of f suffice.

Let Π'_1, Π'_2 be any two arbitrary permutations of $\{0,1\}^k$ and consider any dissection protocol P with respect to (Π'_1, Π'_2). Consider the first meaningful step in the execution of P and suppose that this step was executed by party_1 (the case that the step was executed by party_2 is analogous). This step partitions the total input space $\mathcal{S} = \left\{0, 1, 2, \ldots, 2^k - 1\right\}$ into two *nonempty* subsets, say $I \subset \mathcal{S}$ and $I' = \mathcal{S} \setminus I$ such that $0 \in I$. Let $0 < i < 2^k - 1$ be the *least* integer such that $i \in I$ but $i + 1 \notin I$; such an i must exist since both the sets are non-empty. Consider the rectangles A, B, C and D in Fig. 8(b). We have the following cases.

Case 1: $i \leq 2^{k-1} - 2^{k/2}$. Observe that, for every such i, there exists a level $i + 1$ vertical rectangle of size $2^k - 2i - 1$ that is partitioned into two rectangles, one of which is of size exactly 1. Thus, $\alpha_{\text{worst}} \geq 2^k - 2i + 1 > 2^{(k/2)+1} - 1 > 2^{k/2} - 1$.

Case 2: $2^{k-1} - 2^{k/2} < i < 2^{k-1} - 1$. Observe that, for every such value of i, rectangle A, which is of size 2^{k-1}, is partitioned into two rectangles, of which one is of size at most $2^{k/2}$. Thus, in this case $\alpha_{\text{worst}} \geq \frac{2^k - 1}{2^{k/2}} > 2^{k/2} - 1$.

Case 3: $2^{k-1} - 1 \leq i \leq 2^{k-1} + 1$. In this case, at least one of the rectangles B, C or D is partitioned into two parts one of which is of size at most 2 and thus $\alpha_{\text{worst}} \geq \frac{2^{k-1} - 1}{2} > 2^{k-2} - \frac{1}{2} > 2^{k/2} - 1$.

Case 4: $2^{k-1} + 1 < i < 2^k - 2^{k/2}$. Similar to Case **2**.

Case 5: $i \geq 2^k - 2^{k/2}$. Similar to Case **1**. □

6. Extensions of the Basic Two-party Setup

6.1. Non-tiling Functions

A natural extension of the class of tiling functions involves relaxing the constraint that each monochromatic region *must* be a rectangle.

Definition 12 (δ-tiling function)**.** *A function $f : \{0,1\}^k \times \{0,1\}^k \mapsto \{0,1\}^t$ is a called a δ-tiling function with respect to permutations (Π_1, Π_2) of $\{0,1\}^k$ if each maximal monochromatic region of $A_{\Pi_1,\Pi_2}(f)$ is an union of at most δ disjoint rectangles.*

For example, the function whose tiling is as shown in Fig. 4 is a 2-tiling Boolean function.

Proposition 1. *For any δ-tiling function f with respect to (Π_1, Π_2) with r maximal monochromatic regions, there is a dissection protocol P with respect to (Π_1, Π_2) using at most $4r\delta$ communication steps such that $\alpha_{\mathsf{D}^{\sim\, c}_{u}} \leq (4{+}4c)\,\delta$.*

Proof. We use the algorithm of Theorem 1 on the set of at most $r\delta$ rectangles obtained by partitioning each monochromatic region into rectangles. Since each rectangle is partitioned at most 4 times, each maximal monochromatic region of $A_f(\Pi_1, \Pi_2)$ will be partitioned at most 4δ times. □

6.2. Multi-party Computation

How good is the average Par for a dissection protocol on a d-dimensional tiling function? For a general d, it is non-trivial to compute *precise* bounds because each party_i has her/his own permutation Π_i of the input, the tiles are boxes of *full* dimension and hyperplanes corresponding to each step of the dissection protocol is of dimension *exactly* $d - 1$. Nonetheless, we show that the average Par is very high for dissection protocols even for 3 parties and uniform distribution, thereby suggesting that this quantification of privacy may not provide good bounds for three or more parties.

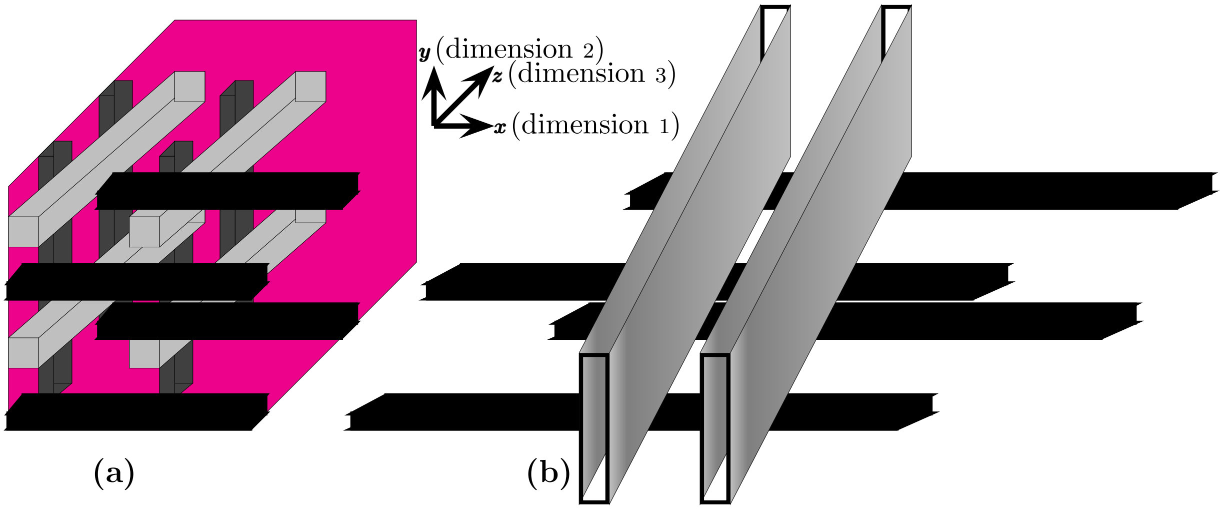


Figure 9: **(not drawn to scale)** **(a)** *The tiling function in the proof of Lemma 3. The non-trivial rectangles for dimensions* 1, 2 *and* 3 *are colored by black, dark gray and light gray, respectively; the trivial rectangles, each having a distinct value, cover the region colored magenta.* **(b)** *Rectangles (in light gray) corresponding to a hypothetically first meaningful step of the protocol.*

Theorem 3 (large average PAR for dissection protocols with 3 parties)**.** *There exists a tiling function $f\colon \{0,1\}^k \times \{0,1\}^k \times \{0,1\}^k \mapsto \{0,1\}^{3k}$ such that, for any three permutations Π_1, Π_2, Π_3 of $\{0,1\}^k$, every dissection protocol with respect to (Π_1, Π_2, Π_3) must have $\alpha_{\mathsf{D}_u} = \Omega\left(2^k\right)$.*

Proof. In the sequel, for convenience *we refer to* 3*-dimensional hyper-rectangles simply by rectangles* and refer to the arguments of function f via *decimal equivalent of the corresponding binary numbers.* The tiling function for this theorem is adopted from an example of the paper by Paterson and Yao [14, 15] with appropriate modifications. The three arguments of f are referred to as dimensions 1, 2 and 3, respectively. Define the *volume* of a rectangle $R = [x_1, x_1'] \times [x_2, x_2'] \times [x_3, x_3'] \subseteq \{0, 1, \ldots, 2^k - 1\}^3$ is $\mathsf{Volume}(R) = \max\{0, \Pi_{i=1}^{3}(x_i' - x_i + 1)\}$. For convenience, let $[*]$ denote the interval $\left[0, 2^k - 1\right]$. We provide the tiling for the function f; see Fig. 9 for a graphical illustration (note that Fig. 9 is *not* drawn to scale):

- For each dimension, we have a set of $\Theta\left(2^{2k}\right)$ rectangles; we refer to these rectangles as *non-trivial* rectangles for this dimension.
  - For dimension 1, these rectangles are of the form $[*] \times [2y, 2y] \times [2z, 2z]$ for every integral value of $0 \leq 2y, 2z < 2^k$.
  - For dimension 2, these rectangles are of the form $[2x, 2x] \times [*] \times [2z+1, 2z+1]$ for every integral value of $0 \leq 2x, 2z+1 < 2^k$.
  - For dimension 3, these rectangles are of the form $[2x+1, 2x+1] \times [2y+1, 2y+1] \times [*]$ for every integral value of $0 \leq 2x+1, 2y+1 < 2^k$.
- The remaining "trivial" rectangles are each of unit volume such that they together cover the remaining input space.

Let $\mathcal{S}_{\text{non-trivial}}$ be the set of all non-trivial rectangles. Observe that:

- Rectangles in $\mathcal{S}_{\text{non-trivial}}$ are mutually disjoint since any two of them do not intersect in at least one dimension.
- *Each* rectangle in $\mathcal{S}_{\text{non-trivial}}$ has a volume of 2^k and thus the sum of their volumes is $\Theta\left(2^{3k}\right)$.

It now also follows that the number of monochromatic regions is $O\left(2^{3k}\right)$. Suppose that a dissection protocol partitions, for $i = 1, 2, \ldots, |\mathcal{S}_{\text{non-trivial}}|$, the

$i^{\rm th}$ non-trivial rectangle $R_i \in \mathcal{S}_{\rm non-trivial}$ into t_i rectangles, say $R_{i,1}, R_{i,2}, \ldots, R_{i,t_i}$. Then,

$$\alpha_{\mathsf{D}_u} \overset{\rm def}{=} \sum_{\substack{(x,y,z)\in\\ \{0,1\}^k\times\{0,1\}^k\times\{0,1\}^k}} \Pr_{\mathsf{D}_u}\left[x \,\&\, y \,\&\, z\right] \frac{\left|R^I(x,y,z)\right|}{\left|R^P(x,y,z)\right|} \geq \sum_{i=1}^{|\mathcal{S}_{\rm non-trivial}|} \sum_{j=1}^{t_i} \sum_{(x,y,z)\in R_{i,j}} \Pr_{\mathsf{D}_u}[x \,\&\, y \,\&\, z] \frac{\mathsf{Volume}\,(R_i)}{\mathsf{Volume}\,(R_{i,j})}$$

$$= \sum_{i=1}^{|\mathcal{S}_{\rm non-trivial}|} \sum_{j=1}^{t_i} \frac{2^k}{2^{3k}} = \sum_{i=1}^{|\mathcal{S}_{\rm non-trivial}|} \left(t_i/2^{2k}\right)$$

Thus, it suffices to show that $\displaystyle\sum_{i=1}^{|\mathcal{S}_{\rm non-trivial}|} t_i = \Omega\left(2^{3k}\right)$. Let $\mathcal{Q}$ be the set of maximal *monochromatic* rectangles produced the partitioning of the entire protocol. Consider the two entries $p_{x,y,z} = (2x+1, 2y, 2z+1)$ and $p'_{x,y,z} = (2x, 2y, 2z)$ (see Fig. 10). Note that $p_{x,y,z}$ belongs to a trivial rectangle since their third, first and second coordinate does not lie within *any* non-trivial rectangle of dimension 1, 2 and 3, respectively, whereas $p'_{x,y,z}$ belongs to the non-trivial rectangle $[*] \times [2 \times (8y), 2 \times (8y)] \times [2 \times (8z), 2 \times (8z)]$ of dimension 1. Thus, $p_{x,y,z}$ and $p'_{x,y,z}$ cannot belong to the same rectangle in $\mathcal{Q}$. Let $T = \bigcup \left\{ \{p_{\,8x,8y,8z},\, p'_{\,8x,8y,8z}\} \,|\, 64 < 16x, 16y, 16z < 2^k - 64 \right\}$. Clearly, $|T| = \Theta\left(2^{3k}\right)$. For an entry (x_1, x_2, x_3), let its neighborhood be defined by the ball $\mathsf{Nbr}(x_1, x_2, x_3) = \left\{ (x'_1, x'_2, x'_3) \,|\, \forall i : |x_i - x'_i| \leq 4 \right\}$, *i.e.*, the neighborhood of an entry is the set of all entries (x'_1, x'_2, x'_3) such that each x'_i lies in the range $\left[x_i - 4, x_i + 4\right]$ for $i = 1, 2, 3$. Note that $\mathsf{Nbr}(p_{\,8x,8y,8z}) \cap \mathsf{Nbr}(p_{\,8x',8y',8z'}) = \emptyset$ provided $(x, y, z) \neq (x', y', z')$. Next, we show that, to ensure that the two entries $p_{\,8x,8y,8z}$ and $p'_{\,8x,8y,8z}$ are in two different rectangles in $\mathcal{Q}$, the protocol must produce an *additional* fragment of one of the non-trivial rectangles in the neighborhood $\mathsf{Nbr}(p_{\,8x,8y,8z})$; this would directly imply $\sum_i t_i = \Omega\left(2^{3k}\right)$.

Consider the step of the protocol *before* which $p_{\,8x,8y,8z}$ and $p'_{\,8x,8y,8z}$ were contained inside the same rectangle, namely a rectangle Q that includes the rectangle $[16x, 16x+1] \times [16y, 16y] \times [16z, 16z+1]$, but after which they are in two different rectangles $Q_1 = [a'_1, b'_1] \times [a'_2, b'_2] \times [a'_3, b'_3]$ and $Q_2 = [a''_1, b''_1] \times [a''_2, b''_2] \times [a''_3, b''_3]$. Remember that both Q_1 and Q_2 must have the same two dimensions and these two dimensions must be the same as the corresponding dimensions of Q. The following cases arise.

Case 1 (split via the first coordinate): $[a'_2, b'_2] = [a''_2, b''_2] \supseteq [16y, 16y]$, $[a'_3, b'_3] = [a''_3, b''_3] \supseteq [16z, 16z+1]$, $b'_1 = 16x$ and $a''_1 = 16x + 1$. Then, a new fragment of a non-trivial rectangle of dimension 2 is produced at $[16x, 16y, 16z] \in \mathsf{Nbr}(p_{8x,8y,8z})$.

Case 2 (split via the second coordinate): $[a'_1, b'_1] = [a''_1, b''_1] \supseteq [16x, 16x + 1]$ and $[a'_3, b'_3] = [a''_3, b''_3] \supseteq [16z, 16z + 1]$. This case is not possible.

Case 3 (split via the third coordinate): $[a'_1, b'_1] = [a''_1, b''_1] \supseteq [16x, 16x + 1]$, $[a'_2, b'_2] = [a''_2, b''_2] \supseteq [16y, 16y]$, $b'_3 = 16z$ and $a''_3 = 16z + 1$. Then, a new fragment of a non-trivial rectangle of dimension 1 is produced at $[16x, 16y, 16z] \in \mathsf{Nbr}(p_{8x,8y,8z})$. ❑

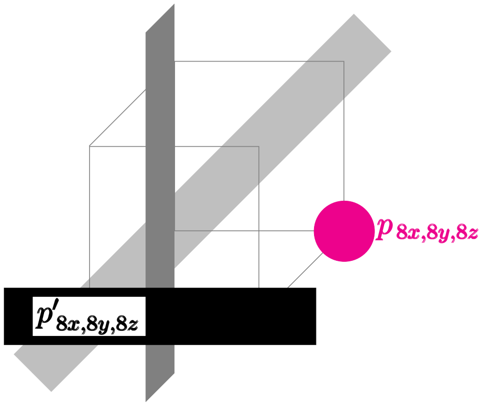


Figure 10: *Separating* $p_{8x,8y,8z}$ *from* $p'_{8x,8y,8z}$.

Remark 2. *A generalized version of the example in d dimension can be used to provide a slightly improved lower bound on α_{D_u} for dissection protocols with more than three parties; the bound asymptotically approaches $\Omega\left(2^{2k}\right)$ for large d.*

7. Analysis of the Bisection Protocol for Two Functions

In Section 5.1 we showed that any Boolean tiling function can be computed with perfect privacy by a dissection protocol. In [8] the authors provided calculated bounds on α_{worst} and α_{D_u} for the *bisection* protocol, a special case of the general dissection protocol (see Definition 9), on a few functions. In this section, we analyze the bisection protocol [9, 10], for two Boolean functions that appear in the literature. As before, D_u denotes the uniform distribution. Letting $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \{0, 1\}^k$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n) \in \{0, 1\}^k$, the functions that we consider are the following:

- $\mathsf{set\text{-}covering}$: $f_{\wedge,\vee}(\mathbf{x}, \mathbf{y}) = \bigwedge_{i=1}^{n} (x_i \vee y_i)$. To interpret this as a set-covering function, suppose that the universe $\mathcal{U}$ consists of n elements $e_1, e_2, \ldots, e_n$ and the vectors $\mathbf{x}$ and $\mathbf{y}$ encode membership of the elements in two sets $S_{\mathbf{x}}$ and $S_{\mathbf{y}}$, *i.e.*, x_i (respectively, y_i) is 1 if and only if $e_i \in S_{\mathbf{x}}$ (respectively, $e_i \in S_{\mathbf{y}}$). Then, $f_{\wedge,\vee}(\mathbf{x}, \mathbf{y}) = 1$ if and only if $S_{\mathbf{x}} \cup S_{\mathbf{y}} = \mathcal{U}$.

equality: $f_=(\mathbf{x},\mathbf{y}) = \begin{cases} 1 & \text{if } \forall\, i:\ x_i = y_i \\ 0 & \text{otherwise} \end{cases}$. The equality function provides a useful testbed for evaluating privacy preserving protocols, *e.g.*, see [3].

As we already noted in Section 2, both of these functions are studied in the context of evaluating privacy preserving protocols and communication complexity settings [3, 11]. A summary of our bounds are as follows.

$f_{\wedge,\vee}$	$\alpha_{\text{worst}} \geq \alpha_{\mathsf{D}_u} \geq \left(\frac{3}{2}\right)^{2k}$	
$f_=$	$\alpha_{\mathsf{D}_u} = 2^k - 2 + 2^{1-k}$	$\alpha_{\text{worst}} = 2^{2k-1} - 2^{k-1}$

We will use the formula for α_{D_u} that we derived in the proof of Theorem 1: *letting r denote the number of monochromatic regions in an ideal partition of the function if, for $i = 1, 2, \ldots, r$, the i^{th} monochromatic region contain $y_i \times 2^{2k}$ elements and the bisection protocol partitions this region into $t_i \geq 1$ rectangles containing $z_1, \ldots, z_{t_i}$ elements, respectively, then $\alpha_{\mathsf{D}_u} = \sum_{i=1}^{r} t_i y_i$.* In the sequel, by "contribution of a rectangle (of the bisection protocol) to the (average PAR)" we mean the size of the ideal monochromatic region that the rectangle is a part.

7.1. Set Covering Function

Theorem 4. $\alpha_{\mathsf{D}_u} \geq (3/2)^{2k}$.

Proof. We begin by showing the geometry of the tilings for small values of k which easily generalizes to larger k. The ideal tiling for $f_{\wedge,\vee}$ is shown in Fig. 11(a) for $k = 3$ with the value of the function for each input pair. The sizes of the ideal monochromatic partition are shown in Fig. 11(b) for $k = 1, 2, 3, 4$. The contributions to the average PAR of various inputs after applying the bisection protocol are illustrated in Fig. 12 for $k = 1, 2, 3, 4$. We observe the following:

- The tiles colored *light gray* for the case when $k = 4$ are referred to as the "background tiles". For $k = 1, 2, 3, 4$ each such tile contributes $3, 9, 27$ and 81, respectively, to the average PAR. In general, this contribution is given by 3^k and all these tiles have size 1.
- The contributions of the tiles in the upper-left region of the matrix are given by the sum of the first $2^k - 1$ natural numbers; thus each of these tiles contribute $2^{2k-1} - 2^{k-1}$.

- For any k, observe that the matrix can be decomposed into 4 quadrants; the following observations can be repeated recursively on each resulting quadrant, except for the first quadrant:
    - The first quadrant is a monochromatic region that contributes $2^{2k-1} - 2^{k-1}$ to the average Par.
    - The fourth quadrant has the same structure as the original matrix, but the contributions for the *non-background* tiles will be related to the case of a matrix with j bits instead of k, where the size of the quadrant is 2^j. For example, notice that the fourth quadrant of a matrix with $k = 4$ is the same as a whole matrix with $k = 3$, except for the "background tiles", that always contribute for 3^k, with the original value of k.
    - The second and third quadrants are similar to the fourth quadrant case, but in this case the values in the upper-left portion of the quadrants will remain the same as the original matrix, instead of going down as with the fourth quadrant case.

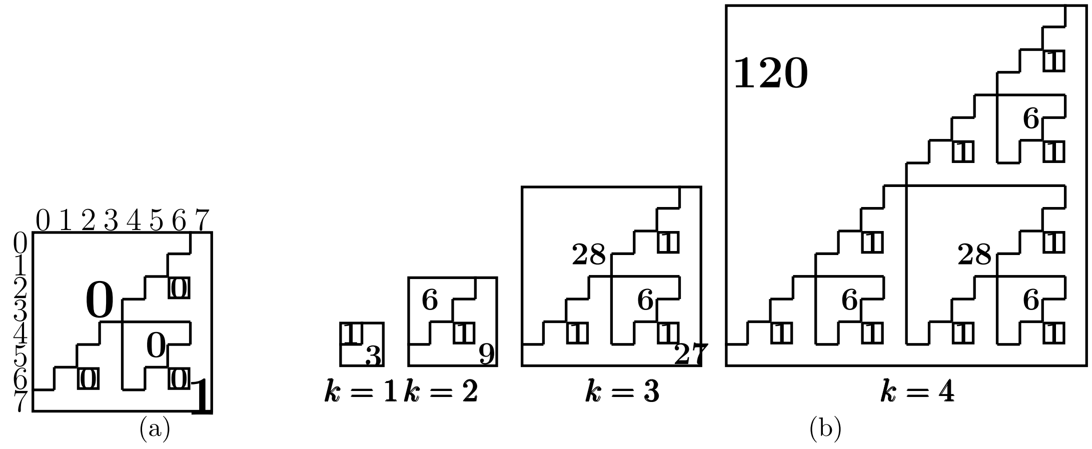
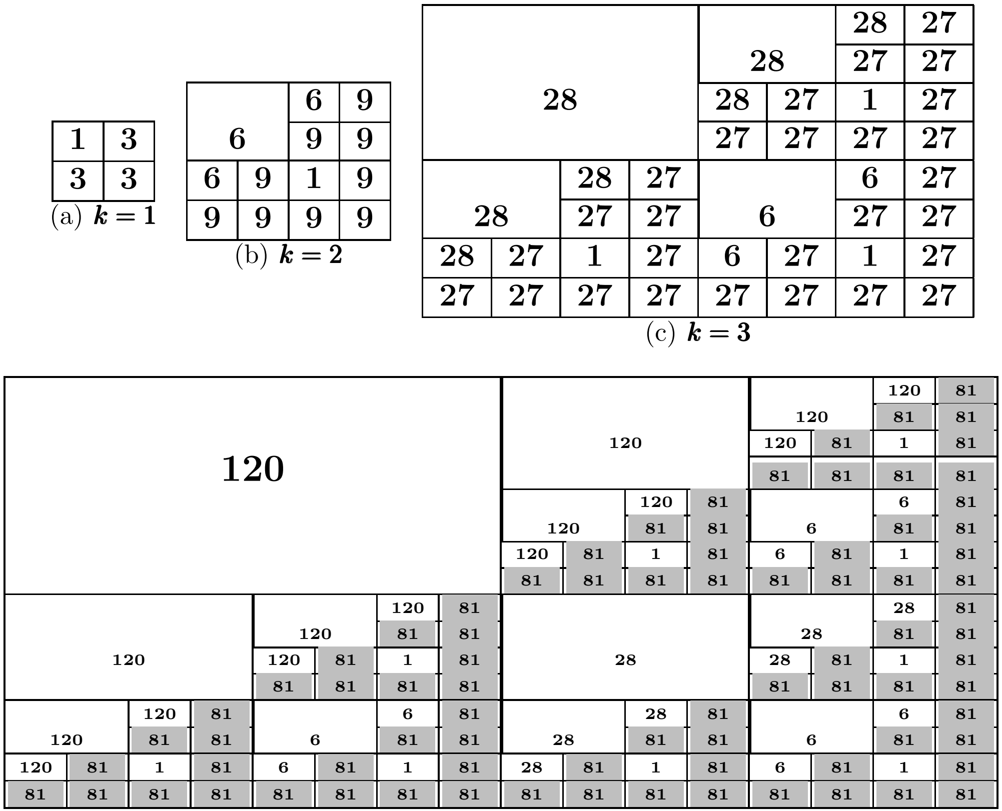


Figure 11: **(a)** Ideal monochromatic partition for $f_{\wedge,\vee}$ when $k = 3$. **(b)** Sizes of ideal monochromatic partition for $f_{\wedge,\vee}$.

Based on these observations, we can obtain a recurrence for the total contribution to the average Par of all the tiles in a generic matrix. We need the following parameters:



(a) $\boldsymbol{k=1}$

(b) $\boldsymbol{k=2}$

(c) $\boldsymbol{k=3}$

(d) $\boldsymbol{k=4}$

Figure 12: Contribution to PAR for $k = 0, 1, 2, 3, 4$.

- The number of bits in the original matrix, that we denote by k;
- The number of bits corresponding to the size of the matrix, or submatrix being considered, that we denote by i;
- The number of bits to be used in the calculation of the contribution of the upper-left portion of the matrix, or submatrix, being considered; we denote this by j.

The recurrence that computes the total contribution to the PAR of all the tiles in the matrix is:

$$g(i, j, k) = \begin{cases} 3^k, & \text{if } i = 0 \\ 2^{2j-1} - 2^{j-1} + 2g(i-1, j, k) + g(i-1, i-1, k), & \text{otherwise} \end{cases}$$

The values of i and j are initially set to the value of k. The interpretation of each term in the above recurrence is as follows:

- 3^k is the contribution of each "background tile";
- $2^{2j-1} - 2^{j-1}$ is the contribution of the first quadrant;
- $g\,(i-1, j, k)$ is the contribution of each one of the second and third quadrants and
- $g\,(i-1, i-1, k)$ is the contribution of the fourth quadrant.

Remember that, for a given k, the recurrence equation is initialized with $i = j = k$. Thus, we have:

Case: $\mathbf{k = 0}$: $g\,(k, k, k) = 3^k = 3^{2k}$.

Case: $\mathbf{k > 0}$: $g(k, k, k) = g(k-1, k-1, k) + 2g(k-1, k, k) + t(k)$. The second parameter to the function indicates how to generate the $t(k)$ terms; the value of such terms is proportional to that parameter. Thus, for $a \geq b$, $g(k, a, k) \geq g(k, b, k)$. For our lower bound, we can neglect the terms $t(k)$. Thus, we obtain:

$$\begin{aligned} g(k, k, k) \geq 3g(k-1, k-1, k) &\geq 3g(k-2, k-2, k) \\ &\geq \cdots\cdots \geq 3g(1, 1, k) \geq 3g(0, 0, k) \end{aligned}$$

For each step, the value of the first parameter decreased exactly by one unit, so after k iterations the value of the first parameter will be zero. Hence we have $g(k, k, k) \geq 3^k g(0, 0, k)$. Since $g(0, 0, k) = 3^k$ we finally obtain $g(k, k, k) \geq 3^k \times 3^k = 3^{2k}$.

Thus, $\alpha_{\mathsf{D}_u} = g(k, k, k)/2^{2k} \geq (3/2)^{2k}$. □

7.2. Equality function

Theorem 5. *$\alpha_{\mathsf{D}_u} = 2^k - 2 + 2^{1-k}$ and $\alpha_{\mathrm{worst}} = 2^{2k-1} - 2^{k-1}$.*

Proof. An illustration of the ideal partition into monochromatic *regions* for equality function is shown in Fig. 13**(a)**. After running the bisection protocol, the induced tiling is (for $k = 3$) is shown in Fig. 13**(b)**. Excluding the diagonal, we have 2 tiles of size 16, 4 tiles of size 4, and 8 tiles of size 1. In

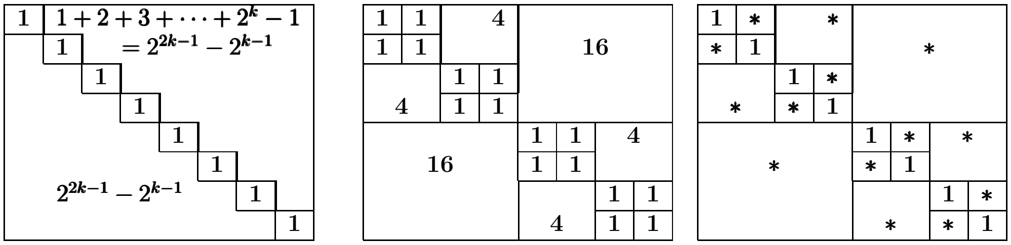


Figure 13: **(a)** Ideal tiling for equality function. **(b)** The induced tiling by the bisection protocol (shown for $k=3$). **(c)** Contribution of each rectangle in protocol-induced tiling where $\ast \equiv \mathbf{2^{2k-1}-2^{k-1}}$. The numbers in the figure denote the size of each tile.

general, it is easy to observe that, for each $0 \le i < k$, we have exactly 2^{k-i} tiles of size 2^{2i}.

The following accounting scheme can be used to simplify calculation. For uniform distribution D_u, α_{D_u} is the sum of the ratio $\frac{|R^I(i,j)|}{|R^P(i,j)|}$ over each element (i,j) in the matrix divided by the number of total elements 2^{2k} in the matrix, where $R^I(i,j)$ and $R^P(i,j)$ is the size of the ideal and protocol-induced tiling that contains the cell (i,j). Consider a rectangle A of size m in the protocol-induced tiling and suppose that A is contained in a monochromatic region of the ideal partition of size m'. Then, the sum of contributions of the elements of A is $\sum_{i=1}^{m} \frac{m'}{m} = m'$. Thus, the total contribution of the rectangle A is simply the size of region of the ideal partition containing it.

Fig. 13**(c)** illustrates the contribution of each rectangle in the protocol-induced tiling to average PAR. We can calculate the total contribution to the average PAR of all the tiles in the matrix, except the diagonal, by multiplying $2^{2k-1}-2^{k-1}$ by the number of tiles. The number of tiles is given by: $\sum_{i=0}^{k-1} 2^{k-i} = 2^{k+1}-2$. The total contribution of those tiles is $(2^{k+1}-2)\times\left(2^{2k-1}-2^{k-1}\right) = 2^{3k}-2^{2k+1}+2^k$. The contribution of the diagonal is $\underbrace{1+1+\cdots\cdots+1}_{2^k \text{ times}} = 2^k$. Since the average objective PAR α_{D_u} is the sum of the total contributions divided by the number of cells in the matrix, we have

$$\alpha_{\mathsf{D}_u} = \frac{2^{3k}-2^{2k+1}+2^k+2^k}{2^{2k}} = \frac{2^{3k}-2^{2k+1}+2^{k+1}}{2^{2k}} = 2^k-2+2^{1-k}$$

It can be seen from the ideal and protocol tilings that the worst case for PAR is the one in which the ideal tile size is $2^{2k-1}-2^{k-1}$, and the protocol tile size is 1. Thus $\alpha_{\text{worst}} = 2^{2k-1} - 2^{k-1}$. □

[1] A. Ghosh, T. Roughgarden and M. Sundararajan. *Universally utility-maximizing privacy mechanisms*, 41th ACM Symposium on Theory of Computing, 351-360, 2009.

[2] P. Berman, B. DasGupta and S. Muthukrishnan. *On the Exact Size of the Binary Space Partitioning of Sets of Isothetic Rectangles with Applications*, SIAM Journal of Discrete Mathematics, 15 (2), 252-267, 2002.

[3] R. Bar-Yehuda, B. Chor, E. Kushilevitz and A. Orlitsky. *Privacy, additional information, and communication*, IEEE Transactions on Information Theory, 39, 55-65, 1993.

[4] F. d'Amore and P. G. Franciosa. *On the optimal binary plane partition for sets of isothetic rectangles*, Information Processing Letters, 44, 255-259, 1992.

[5] D. Chaum, C. Crépeau and I. Damgaard. *Multiparty, unconditionally secure protocols*, 22th ACM Symposium on Theory of Computing, 11-19, 1988.

[6] B. Chor and E. Kushilevitz. *A zero-one law for boolean privacy*, SIAM Journal of Discrete Mathematics, 4, 36-47, 1991.

[7] C. Dwork. *Differential privacy*, 33rd International Colloquium on Automata, Languages and Programming, 1-12, 2006.

[8] J. Feigenbaum, A. Jaggard and M. Schapira. *Approximate Privacy: Foundations and Quantification*, ACM Conference on Electronic Commerce, 167-178, 2010.

[9] E. Grigorievaa, P. J.-J. Heringsb, R. Müllera and D. Vermeulena. *The communication complexity of private value single-item auctions*, Operations Research Letters, 34, 491-498, 2006.

[10] E. Grigorievaa, P. J.-J. Heringsb, R. Müllera and D. Vermeulena. *The private value single item bisection auction*, Economic Theory, 30, 107-118, 2007.

[11] E. Kushilevitz and N. Nisan. *Communication Complexity*, Cambridge University Press, 1997.

[12] D. Kifer and B.-R. Lin. *An Axiomatic View of Statistical Privacy and Utility*, to appear in Journal of Privacy and Confidentiality (conference version appeared in 2010 ACM SIGMOD/PODS Conference).

[13] E. Kushilevitz. *Privacy and communication complexity*, SIAM Journal of Discrete Mathematics, 5 (2), 273-284, 1992.

[14] M. Paterson and F. F. Yao. *Efficient binary space partitions for hidden-surface removal and solid modeling*, Discrete and Computational Geometry, 5(1), 485-503, 1990.

[15] M. Paterson and F. F. Yao. *Optimal binary space partitions for orthogonal objects*, Journal of Algorithms, 13, 99-113, 1992.

[16] C. D. Tóth. *Binary Space Partitions: Recent Developments*, in Combinatorial and Computational Geometry, J. E. Goodman, J. Pach and E. Welzl (eds.), MSRI Publications, 52, 529-556, Cambridge University Press, 2005.

[17] A. C. Yao. *Some complexity questions related to distributive computing*, 11th ACM Symposium on Theory of Computing, 209-213, 1979.