Chapter 10

A Review of Some Approximate Privacy Measures of Multi-Agent Communication Protocols

Bhaskar DasGupta and Venkatakumar Srinivasan

Abstract Privacy preserving computational models have become an important research area due to the increasingly widespread usage of sensitive data in networked environments, as evidenced by distributed computing applications and game-theoretic settings (e.g., auctions). 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. In this chapter, we review a few such notions and known results for them.

10.1 Introduction

Consider the following standard two-agent communication model as originally proposed by Yao (1979). We have two agents, say Alice and Bob, interacting via a public channel as depicted in 10.1. Each of Alice and Bob holds a private input, $x_1 \in \mathbb{X}_1$ and $x_2 \in \mathbb{X}_2$ respectively, that is known to her and him, respectively, and they would like to compute a function $f : \mathbb{X}_1 \times \mathbb{X}_2 \mapsto \mathbb{R}$ of their two private inputs. Alice and Bob alternately execute "rounds" of a "communication protocol", where in each round they make available a small amount of information about their private inputs, such as an answer to a range query on their private inputs or a few bits of their private inputs, until each of them has seen enough information to be able to compute the value of $f(x_1, x_2)$. This setting can be generalized in an obvious manner to $d > 2$ agents computing a d-ary function $f : \mathbb{X}_1 \times \mathbb{X}_2 \times \cdots \times \mathbb{X}_d \mapsto \mathbb{R}$ by allowing each agent to broadcast information about its private input via a public communication channel in a round-robin order; see 11.2 for an illustration. Without loss of generality and to simplify exposition, we may assume that $\mathbb{X}_1 = \mathbb{X}_2 = \cdots = \mathbb{X}_d = \{0, 1, 2, \ldots, n\}$ for some positive integer n that is a power of two. With this assumption, the function f can also be visualized as $n \times n \times \cdots \times n$ d-dimensional real matrix $A_{f,d}$ in which the i^{th} dimension represents the possible inputs of the i^{th} agent, and each entry contains the value of f associated with a particular set of inputs from the d agents (i.e., $A_{f,d}[x_1, x_2, \ldots, x_d] = f(x_1, x_2, \ldots, x_d)$). We will denote $A_{f,2}$ simply as A_f.

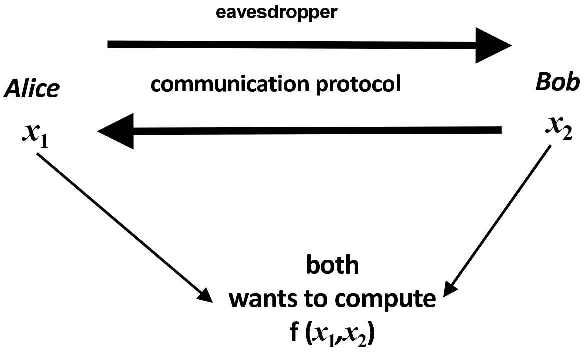


Fig. 10.1 The standard two-agent communication model [Yao (1979)].

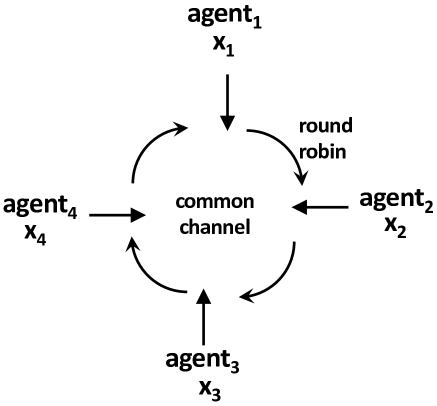


Fig. 10.2 The d-agent communication model for $d = 4$ computing a function $f(x_1, x_2, x_3, x_4)$.

A typical line of research in the above two and multi-agent model of communication, starting with the seminal work of Yao (1979), lies in investigating the *communication complexity* issues, e.g., how many rounds of communications are necessary or sufficient to compute a given family of functions. Such investigations have resulted in a well-developed research area with many interesting results; the reader is referred to a textbook such as Kushilevitz and Nisan (1997) for an overview and basic results in this field, such as the number of bits that need to be exchanged in a two-agent communication protocol by Alice and Bob to compute a given function f is at least $\log_2(\text{rank}_f)$ where rank_f is the linear rank of the matrix A_f over the reals [Kushilevitz and Nisan (1997)] Applications of these results and techniques have led to the famous AT^2 bound for VLSI networks and analysis of dynamic data structures, to mention a few.

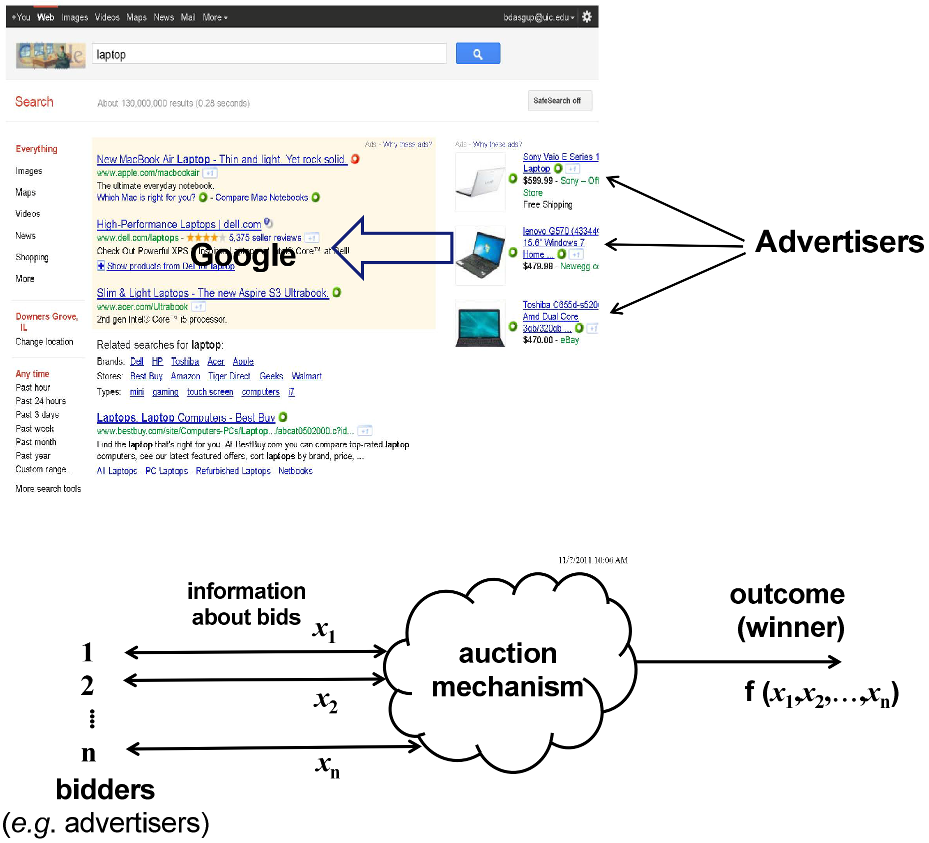
The central question that is the topic of the paper is however motivated differently from the previous line of research by recent increasingly widespread usage of sensitive data in networked environments, as evidenced by distributed computing applications, game-theoretic settings (e.g., auctions) and more. For example, one motivation, as explained in details in Feigenbaum *et al.* (2010), comes from privacy concerns in auction theory in Economics. An offline or online auction can obviously be viewed as an exchange of information between the bidders and the auctioneer where the goal is to the compute the function that determines the winner of the auction (see 11.3). Traditionally desired goals of designing auction mechanisms include maximizing revenues and ensuring that the designed mechanisms are indeed truthful, i.e., bidders fare best by letting their truthful bids known [Nisan *et al.* (2007)]. However, more recently, another complementary goal that has gained significant attention, specially in the context of online auctions such as administered by google and other similar companies, is to be able to preserve privacy of the bidders, i.e., bidders must not reveal more information that what is necessary to the auctioneer for optimal outcomes [Comi *et al.* (2012); Feigenbaum *et al.* (2010)]. Thus, for these types of multi-agent communication protocols, the alternate goal is to preserve the privacy of the agents as opposed to optimizing the communication complexity. Informally, the privacy question that is the focus of this chapter is the following: *given a communication protocol to compute a function via multi-agent communication, how can we quantify the amount of extra information about the agents private inputs, beyond what is necessary to compute the function value, that is revealed by the execution of the protocol*? Note that there are two conflicting constraints: the agents do need to communicate sufficient information for computing the function value, but would prefer not to communicate too much information about their private inputs.

To give a concrete example, consider a sequential second-price auction[1] of an item via a protocol in which the price of the item is incrementally increased and publicly announced until the winner is determined. However, such a protocol publicly reveals more information about the bidders than what is absolutely necessary to determine the winner which could be detrimental for the bidders. For example, the protocol reveals 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 chapter, we will review a generalized geometric privacy framework that captures applications of the above type as well as other applications in multi-agent computation.

10.1.1 *Perfect vs. Approximately perfect privacy*

Unfortunately, even though perfect privacy is the most desirable goal, it is often either impossible, or infeasibly costly to achieve (e.g., requiring impractically extensive communication steps). For example, using the combinatorial characterization of privately computable functions put forth by Chor and Kushilevitz (1991) and Kushilevitz (1992), it is possible to show that the millionaires problem (defined in

[1] In such an auction, the winner is the bidder with the highest bid and the price paid by the winner is that of the second-highest bid.



+You Web Images Videos Maps News Mail More ▾ bdasgup@uic.edu ▾

laptop

Search About 130,000,000 results (0.28 seconds) SafeSearch off

Everything
Images
Maps
Videos
News
Shopping
More

Downers Grove, IL
Change location

Any time
Past hour
Past 24 hours
Past 3 days
Past week
Past month
Past year
Custom range...

More search tools

Ads - Why these ads?

New MacBook Air Laptop - Thin and light. Yet rock solid.
www.apple.com/macbookair +1
The ultimate everyday notebook.
Which Mac is right for you? - Compare Mac Notebooks

High-Performance Laptops | dell.com
www.dell.com/laptops - ★★★★☆ 5,375 seller reviews +1
Check Out Powerful XPS & Inspiron Laptops with Intel® Core™ at Dell!
Show products from Dell for laptop

Slim & Light Laptops - The new Aspire S3 Ultrabook.
www.acer.com/Ultrabook +1
2nd gen Intel® Core™ i5 processor.

Related searches for **laptop**:
Brands: Dell HP Toshiba Acer Apple
Stores: Best Buy Amazon Tiger Direct Geeks Walmart
Types: mini gaming touch screen computers i7

Laptops: Laptop Computers - Best Buy
www.bestbuy.com/site/Computers-PCs/Laptop.../abcat0502000.c?id... +1
Find the **laptop** that's right for you. At BestBuy.com you can compare top-rated **laptop** computers, see our latest featured offers, sort **laptops** by brand, price, ...
All Laptops - PC Laptops - Refurbished Laptops - Netbooks

Ads - Why these ads?

Sony Vaio E Series 1
Laptop +1
$599.99 - Sony – Off
Store
Free Shipping

lenovo G570 (43344
15.6" Windows 7
Home ... +1
$479.99 - Newegg.c

Toshiba C655d-s520
Amd Dual Core
3gb/320gb ... +1
$470.00 - eBay

Fig. 10.3 An online auction mechanism viewed as a multi-agent communication problem.

Section 10.3) is not perfectly privately computable [Chor and Kushilevitz (1991)] and the two-bidder second-price Vickrey auction (also defined in Section 10.3) is perfectly privately computable but only at the cost of and exponential amount of communication by the bidders [Brandt and Sandholm (2008)]. Thus, much of the current research works focus on quantifying and analyzing approximate notions of privacy [Ada *et al.* (2012); Comi *et al.* (2012); Dwork (2006); Feigenbaum *et al.* (2010)].

10.1.2 *Privacy analysis in other environments*

Besides the distributed function computation environment, privacy preserving methods have also been studied in other environments. For example, in the context of mining statistical databases a privacy preserving protocol is expected to provide reliable information about a record being queried while revealing as little information as possible about other records in the database. One model to address the privacy issue in such a data mining environment is the so-called *Differential Privacy Model* originally introduced by Dwork (2006). This model was introduced to investigate the issue of preserving privacy in statistical databases. For example, if the statistical database is a representative sample of some population, differential privacy model allows a user to learn the properties of the population while preserving the privacy of individuals in the population. More formally, a randomized query returning algorithm $\mathcal{A}$ provides ε-differential privacy if, for all pairs of data sets $\mathcal{D}_1$ and $\mathcal{D}_2$ differing on at most one element and for all subset $\mathcal{S}$ of answers provided by $\mathcal{A}$, we have[2]

$$\begin{aligned}&\Pr\left[\text{querying on } D_1 \text{ returns a member in } \mathcal{S}\right]\\ &\qquad\qquad \leq \mathrm{e}^{\varepsilon}\Pr\left[\text{querying on } D_2 \text{ returns a member in } \mathcal{S}\right]\end{aligned}$$

where the probabilities are taken is over the coin tosses of the randomized algorithm $\mathcal{A}$ (i.e., informally, removing one record from the database does not make a query output too much more or less likely). Differential privacy is usually achieved by computing the correct answer to a query and adding a noise drawn from the so-called $\mathcal{L}aplace(f(\varepsilon))$ distribution for some appropriate function f. This approach is sufficient to handle individual queries. In Dwork (2006), the author also provides a mechanism for ensuring differential privacy in case of adaptive queries.

Alternative frameworks, such as the k-anonymization approach [Sweeney (2002)], has also been used for this application.

10.2 Various Frameworks to Quantify Privacy of Protocols

In this section, we review several well-known frameworks to quantify exact or approximate privacy of communication protocols for distributed function computation, one of which is the main topic of this chapter.

10.2.1 *Communication complexity based approaches*

The origin of these frameworks can be traced back to the early works of Chor and Kushilevitz (1991) on characterizations of privately computable functions that can be computed in a perfect private manner, and that of Kushilevitz (1992) on communication complexity issues of privately computable functions. Based on these results, the following two privacy frameworks were independently developed by researchers (an exact characterization of the relationship between these two frameworks is still an open research question):

- Bar-Yehuda *et al.* (1993) provided a combinatorial framework to quantify the amount of privacy that can be maintained in the computation of a function, and the communication cost of achieving this amount of privacy.

[2] e denotes the base of natural logarithm.

- Geometric frameworks to quantify exact and approximate privacy in computing a function in two- and multi-agent communication settings were first formulated by Feigenbaum *et al.* (2010) and subsequently further analyzed by Comi *et al.* (2012). This geometric framework is the main framework that is reviewed in this chapter.

Notable among other results on this approach is the work of Brandt and Sandholm (2008) that, using the framework of Kushilevitz (1992), provided an exponential lower bound on the communication complexity of privacy-preserving second-price Vickrey auctions.

10.2.2 *Information-theoretic approaches*

The study of information-theoretically private protocols can be traced back to the works in Ben-Or *et al.* (1988); Chaum, Crepeau and Damgaard (1988). An underlying assumption in these works was that a constant fraction of the agents are honest, i.e., these agents follow the protocol even if deviating from the protocol may benefit them.

10.2.3 *Cryptographic approaches*

Another approach to securing privacy in any multi-agent communication is to use cryptographic tools that rely on various (mostly unproven but always widely believed to be true) complexity-theoretic assumptions. The origin of this line of research can be traced back to the earlier works of Yao [Yao (1982, 1986)]. Usually these types of protocols are very communication intensive, though communication efficient cryptographic protocols have indeed been obtained in some recent papers in mechanism design problems [Dodis *et al.* (2000); Naor *et al.* (1999)].

10.2.4 *Two-agent differential privacy framework*

This framework, introduced by McGregor *et al.* (2010), attempts to extend the differential privacy model mentioned in Section 10.1.2 in the context of distributed function computation in a two-agent communication setting. In this setting the two agents, say agents A and B, want to find out the hamming distance between the n bit inputs that they hold. This setting is defined in the following manner:

- A *mechanism* M (on Σ^n) is a family of probability distributions $\{\mu_x : x \text{ is an input value}\}$ on $\mathbb{R}$. Such a mechanism M is ε-differentially private if and only if the following condition holds:
  - $\forall\, x, x \in \Sigma^n : \left|x - x\right|_H = 1$, and
  - for all measurable subsets S of $\mathbb{R}$, $\mu_x(S) \leq \mathrm{e}^{\varepsilon}\mu_x(S)$.

  where $\left|x - x\right|_H$ denotes the Hamming distance between x and x'.

- $\mathsf{VIEW}^A_{\mathcal{P}}(x,y)$ is the joint probability distribution over inputs x, y, the transcript of a protocol $\mathcal{P}$ and the private randomness of agent A (the probability space is private randomness for both agents). $\mathsf{VIEW}^B_{\mathcal{P}}(x,y)$ is defined in a similar manner with respect to agent B.

Then, a communication protocol $\mathcal{P}$ has ε-differential privacy if and only if both of the following conditions hold:

(a) For all input x, $\mathsf{VIEW}^A_{\mathcal{P}}(x,y)$ is ε-differential private.
(b) For all input y, $\mathsf{VIEW}^B_{\mathcal{P}}(x,y)$ is ε-differential private.

A major contrition of McGregor *et al.* (2010) is a *lower bound* on the *least additive error* of any differentially private protocol that is used to compute the hamming distance.

Theorem 10.1. [McGregor *et al.* (2010)] *Let $\mathcal{P}(x,y)$ be a randomized protocol with ε-differential privacy for inputs $x, y \in \left\{0,1\right\}^n$, and let $\delta > 0$. Then, with probability at least $1-\delta$ over $x, y \in \{0,1\}^n$ and the coin tosses of $\mathcal{P}$, output of agent B differs from $\langle x, y\rangle$ by at least $\Omega\left(\frac{\sqrt{n}}{\log n} \times \frac{\delta}{e^{\varepsilon}}\right)$.*

An obvious research question for investigation is to see if the above lower bound can be improved or if an actual protocol with a matching upper bound can be found.

10.3 Benchmark Problems and Functions

Often the usefulness of a privacy definition in distributed function computation is checked by demonstrating its value for a class of interesting functions ("benchmark" functions). We mention a few such functions here.

Set-covering function $f_{\textbf{set-cover}}$: Suppose that the universe $\mathcal{U}$ consists of k elements $u_1, u_2, \ldots, u_k$, and the vectors $\vec{x} = (x_1, x_2, \ldots, x_k) \in \{0,1\}^k$ and $\vec{y} = (y_1, y_2, \ldots, y_k) \in \{0,1\}^k$ encode membership of the elements in two sets $S_{\vec{x}}$ and $S_{\vec{y}}$, i.e., x_i (respectively, y_i) is 1 if and only if $u_i \in S_{\vec{x}}$ (respectively, $u_i \in S_{\vec{y}}$). Then,

$$f_{\text{set-cover}}(\vec{x},\vec{y}) \stackrel{\text{def}}{=} \bigwedge_{i=1}^{k} (x_i \vee y_i) = \begin{cases} 1, & \text{if } S_{\vec{x}} \cup S_{\vec{y}} = \mathcal{U} \\ 0, & \text{otherwise} \end{cases}$$

Set-covering type of functions are useful for studying the differences between deterministic and non-deterministic communication complexities [Kushilevitz and Nisan (1997)].

Equality function $f_=$: For two boolean vectors $\vec{x} = (x_1, x_2, \ldots, x_k) \in \{0,1\}^k$ and $\vec{y} = (y_1, y_2, \ldots, y_k) \in \{0,1\}^k$:

$$f_=(\vec{x},\vec{y}) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if } \forall\, i\colon\ x_i = y_i \\ 0, & \text{otherwise} \end{cases}$$

The equality function provides a useful testbed for evaluating privacy preserving protocols [Bar-Yehuda *et al.* (1993); Kushilevitz and Nisan (1997)]

Set-disjointness function $f_{\textbf{disjoint}}$: We have two boolean vectors $\vec{x} = (x_1, x_2, \ldots, x_k) \in \{0,1\}^k$ and $\vec{y} = (y_1, y_2, \ldots, y_k) \in \{0,1\}^k$ (encoding set memberships of elements from an universe) and we wish to decide if they disagree on every coordinate or not, i.e.,

$$f_{\text{disjoint}}\left(\vec{x}, \vec{y}\right) \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if } \forall\, i\colon\ x_i \neq y_i \\ 0, & \text{otherwise} \end{cases}$$

The set-disjointness function plays an important role in the theory and application of communication complexity.

Set-intersection function $f_{\textbf{intersect}}$: We have two boolean vectors $\vec{x} = (x_1, x_2, \ldots, x_k) \in \{0,1\}^k$ and $\vec{y} = (y_1, y_2, \ldots, y_k) \in \{0,1\}^k$ (encoding set memberships of elements from an universe) and we wish to determine the coordinates in which both of them have a 1, i.e.,

$$f_{\text{intersect}}\left(\vec{x}, \vec{y}\right) \stackrel{\text{def}}{=} \left(z_1, z_2, \ldots, z_n\right) \text{ where, for each } 1 \leq j \leq k,\ z_j = x_j \wedge y_j$$

The set-intersection function has motivated the study of privacy-preserving computation for many years. A typical application of the set-intersection function is when two organizations wish to compute the set of common members without disclosing the members of only one of the organizations [Feigenbaum *et al.* (2010)].

Millionaires problem $f_{\textbf{millionaire}}$: In this problem, the two agents are two millionaires, each knowing his/her own wealth as own private information, and the goal is to discover the identity of the richer millionaire while preserving the privacy of both agents. Formally, for two boolean vectors $\vec{x} = (x_0, x_1, \ldots, x_{k-1}) \in \{0,1\}^k$ and $\vec{y} = (y_0, y_1, \ldots, y_{k-1}) \in \{0,1\}^k$:

$$f_{\text{millionaire}}\left(\vec{x}, \vec{y}\right) \stackrel{\text{def}}{=} \begin{cases} 0, & \text{if } \sum\limits_{j=0}^{k-1} 2^j\, x_j \geq \sum\limits_{j=0}^{k-1} 2^j\, y_j \\ 1, & \text{otherwise} \end{cases}$$

Privacy-preserving protocols for the millionaires problem was investigated in papers such as Chor and Kushilevitz (1991); Feigenbaum *et al.* (2010); Yao (1979).

Second-price Vickrey auction: In a 2nd-price Vickrey auction [Vickrey (1961)] involving one item and two bidders, each having a private value of the item, the goal is to declare the bidder with the higher value as the winner (breaking ties arbitrarily) and reveal the identity of the winner as well as the value of the losing bidder. Formally, for two boolean vectors $\vec{x} = (x_0, x_1, \ldots, x_{k-1}) \in \{0,1\}^k$ and $\vec{y} = (y_0, y_1, \ldots, y_{k-1}) \in \{0,1\}^k$:

$$f_{\text{Second-priceVickrey}}\left(\vec{x}, \vec{y}\right) \stackrel{\text{def}}{=} \begin{cases} \left(0, y_0, y_1, \ldots, y_{k-1}\right), & \text{if } \sum\limits_{j=0}^{k-1} 2^j\, x_j \geq \sum\limits_{j=0}^{k-1} 2^j\, y_j \\ \left(1, x_0, x_1, \ldots, x_{k-1}\right), & \text{otherwise} \end{cases}$$

Second-price Vickrey auction is a fundamental technique in mechanism design for inducing truthful behavior in one-item auctions [Nisan *et al.* (2007)].

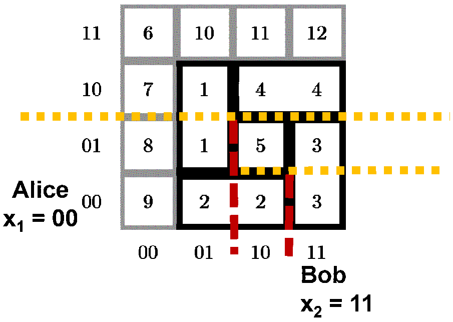


Fig. 10.4 Illustration of one run of the bisection protocol when Alice and Bob has private inputs 00 and 11, respectively. The recursive partitioning induced by the execution of the protocol is shown by thick dashed lines.

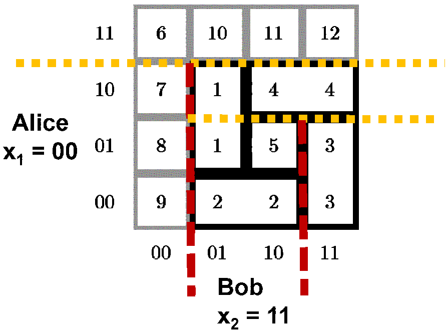


Fig. 10.5 Illustration of one run of the dissection protocol when Alice and Bob has private inputs 00 and 11, respectively. The recursive partitioning induced by the execution of the protocol is shown by thick dashed lines.

10.4 Examples of Standard Communication Protocols

A protocol $\mathcal{P}$ will refer to (a priori) fixed set of rules for communication, and the transcript of $\mathcal{P}$ is the total information (e.g., bits) exchanged during an execution of $\mathcal{P}$. By a "run" of the protocol, one refers to the entire execution of the protocol for a specific set of private inputs of the two agents. For simplicity, we illustrate these protocols for two agents only, but they are easily generalized for the case of $d > 2$ agents in an obvious manner. Typically, one assumes that in each communication round of a protocol $\mathcal{P}$, except the very last one, one of the agents alternately sends out a bit that is computed as a function of that agents' input and communication history. The last message sent in $\mathcal{P}$ is assumed to contain the actual value of the function and therefore may require a larger number of bits. The final outcome of the protocol $\mathcal{P}$ is denoted by the function $t_{\mathcal{P}}$. Viewed in this manner, each run of a protocol recursively induces a partition of the associated matrix A_f of a function f. Three such well-known communication protocols studied in the literature are as follows:

α-bisection protocol: For a constant $\alpha \in \left[\frac{1}{2}, 1\right)$, a protocol is a α-bisection protocol provided the following two conditions hold:

- at each communication step, the maintained subset of inputs of each agent is a set of contiguous integers, and
- at each communication step, the communicating agent partitions its input space of size z into two halves of size $\alpha\, z$ and $(1-\alpha)\, z$.

Bisection protocol: A $\frac{1}{2}$-bisection protocol is simply called a bisection protocol (see 10.4 for an illustration).

Bounded-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 agent repeatedly partitions its input space into two halves one of which is of size exactly one.

Sealed-bid auction protocol: This straightforward protocol is applicable for functions that represent the auction of an item. Here the auctioneer receives sealed bids from all bidders and computes the outcome based on this information.

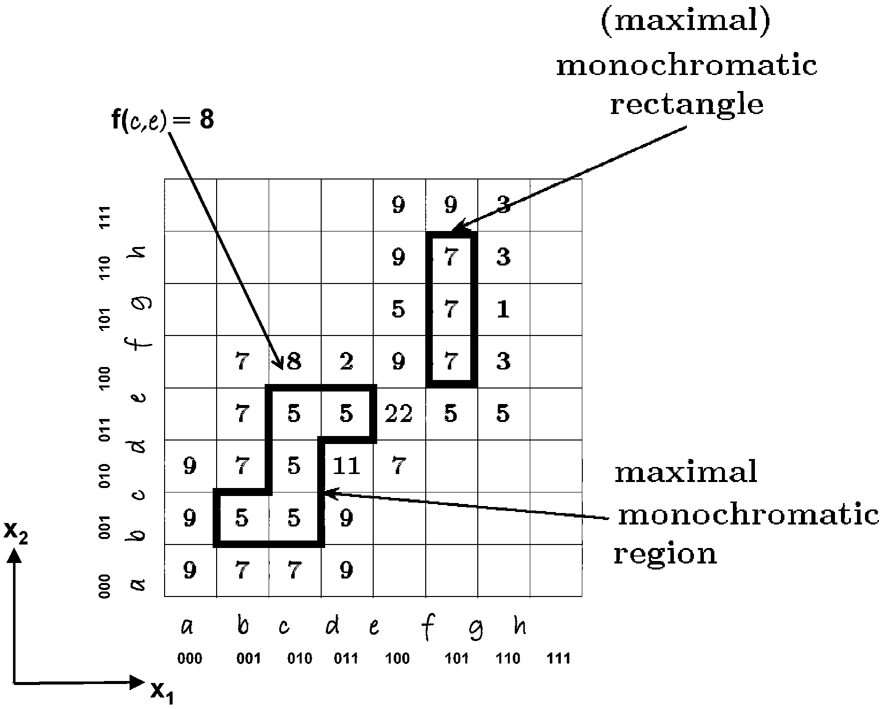
Ascending-price English auction protocol: This straightforward protocol is applicable for functions that represent the auction of an item. Suppose that we have two bidders only. We start with a price of zero for the item and, in each discrete time step increment, we increase the price by one until one of the two bidders indicates that his/her value for the item is less than the current price, say a. Then, we allocate the item to the other bidder for a price of $a - 1$.

Comi et al. [Comi *et al.* (2012)] considered a more general version of the α-bisection protocol in the following manner. When designing protocols for environments in which the input of each agent has a natural ordering (e.g., the set of input of an agent from $\left\{0,1\right\}^k$ can represent the numbers $0, 1, 2, \ldots, 2^k - 1$, as is in the case when computing the maximum/minimum of two inputs, in the millionaires problem, in second-price auctions, and more), a natural restriction is to allow protocols such that each agent asks questions of the form "*Is your input between a and b (in this natural order over possible inputs)?*", where $a, b \in \left\{0,1\right\}^k$. Notice that such a protocol divides the input space into two (not necessarily equal) halves (see 10.5). Such protocols were termed as the dissection protocol in Comi *et al.* (2012) and were useful in analyzing average loss of privacy.

10.5 A Geometric Approach to Quantify Privacy

In this section, we review a recent interesting geometric approach to privacy is based on communication complexity that was initiated by Feigenbaum et al. [Feigenbaum *et al.* (2010)] and subsequently followed up in Ada *et al.* (2012); Comi *et al.* (2012). Although originally motivated by agents' privacy in mechanism design, the definitions and tools can be easily applied to distributed function computation in general. This framework allows one to quantify approximate privacy as well as study the trade-off between privacy preservation and communication complexity. For simplicity of exposition, we discuss the framework first for two agents and later comment on how to generalize it when $d > 2$ agents communicate. As mentioned in the introduction, we have two agents and a function $f \colon \mathbb{X} \times \mathbb{X} \mapsto \mathbb{R}$ of two arguments to compute, where $\mathbb{X} = \{0, 1, 2, \ldots, n\}$ with $n = 2^k$ for some positive integer k, and such a function f can be visualized via the associated two-dimensional matrix A_f. For convenience, we will view the elements in $\mathbb{X}$ in binary as a k-bit number whenever required.

Intuitively, a quantification of (exact or approximate) privacy should satisfy the objective that any observer of the protocol $\mathcal{P}$ should not be able to distinguish the private inputs of the two communicating agents from as large a set as possible of other possible private inputs. To capture this intuition, Feigenbaum et al. [Feigenbaum *et al.* (2010)] makes use of the machinery of communication-complexity theory to introduce the so-called *Privacy Approximation Ratio* (Par) via a geometric and combinatorial interpretation of protocols. To define Par, we first need to state some basic communication complexity definitions for a two-agent communication model [Kushilevitz and Nisan (1997)].



	000	001	010	011	100	101	110	111
111					9	9	3	
110 h					9	7	3	
101 g					5	7	1	
100 f		7	8	2	9	7	3	
011 e		7	5	5	22	5	5	
010 d	9	7	5	11	7			
001 c b	9	5	5	9				
000 a	9	7	7	9				

Fig. 10.6 Illustration of concepts in Definition 10.1.

Definition 10.1 (see 11.5 for an illustration).

(I) A region R of A_f is any subset of entries in A_f. R is monochromatic if all entries in R are of the same value. A monochromatic region R of A_f is maximal if no other monochromatic region in A_f properly contains it. The ideal monochromatic partition of A_f is made up of the maximal monochromatic regions.

(II) A partition of A_f is a collection of disjoint regions in A_f whose union equals

to A_f. A monochromatic partition of A_f is a partition all of whose regions are monochromatic.

(III) *A rectangle in A_f is a sub-matrix of A_f. A tiling of A_f is a partition of A_f into rectangles. A tiling T_1 of A_f is said to be a refinement of another tiling T_2 of A_f if every rectangle in T_1 is contained in some rectangle in T_2.*

(IV) *A protocol $\mathcal{P}$ achieves perfect privacy if, for every pair 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 $t_{\mathcal{P}}(x_1,\ x_2) = t_{\mathcal{P}}(x'{}_1,\ x'{}_2)$.*

(V) *A communication protocol $\mathcal{P}$ for f is perfectly privacy-preserving if the monochromatic tiling induced by P is the ideal monochromatic partition of A_f.*

(VI) *Let $R^P(x_1, x_2)$ be the monochromatic rectangle induced by protocol P for $(x_1, x_2) \in \{0,1\}^k \times \{0,1\}^k$ and $R^I(x_1, x_2)$ be the monochromatic region containing $A_f[x_1, y_1]$ in the ideal monochromatic partition of A_f. Then P has a worst-case privacy-approximation-ratio (*Par*) of Δ_{worst} if*

$$\Delta_{\text{worst}} = \max_{(x_1,x_2)} \left[\frac{|\, R^I(x_1, x_2)\,|}{|\, R^P(x_1, x_2)|} \right]$$

See 10.7 for an illustration.

(VII) *Let $\mathcal{D}$ be a probability distribution over the space of inputs. The average case privacy-approximation-ratio (*Par*) of a communication protocol P under distribution $\mathcal{D}$ for function f is*

$$\Delta_{\mathcal{D}} = \mathbb{E}_{\mathcal{D}} \left[\frac{|R^I(x_1, x_2)|}{|R^P(x_1, x_2)|} \right]$$

where $\mathbb{E}_{\mathcal{D}}$ denotes the expectation with respect to the distribution $\mathcal{D}$.

(VIII) *The worst case* Par *for a function f is the minimum, over all protocols P for f, of the worst case* Par *of P.*

In Definition 10.1(VI)–(VIII), the underlying assumption is that partitioning an ideal monochromatic rectangle results in loss of privacy. The intuition behind this is as follows. Consider the situation depicted in 10.8 where the shaded ideal monochromatic rectangle is partitioned into two rectangles by a protocol. Note that the value of $f(x, y)$ is the same for all $x_1 \leq x \leq x_2$ and $y_1 \leq y \leq y_2$ since the shaded rectangle is monochromatic. But, observing the protocol allows one to distinguish between subsets of these inputs, namely inputs in the subset $\{\,(x, y)\,|\, x_1 \leq x \leq x_2, y_1 \leq y < y'\}$ from inputs in the subset $\{\,(x, y)\,|\, x_1 \leq x \leq x_2, y' \leq y < y_2\}$, thereby revealing extra information.

Using the above framework and definitions, Feigenbaum et al. [Feigenbaum *et al.* (2010)] provided calculations of worst-case and average Par values for a number of functions as summarized in Table 10.1.

10.5.1 *Tiling functions and dissection protocols*

Comi et al. [Comi *et al.* (2012)] further investigated this geometric approach by defining a special class of functions called the "tiling" functions, and analyzing the

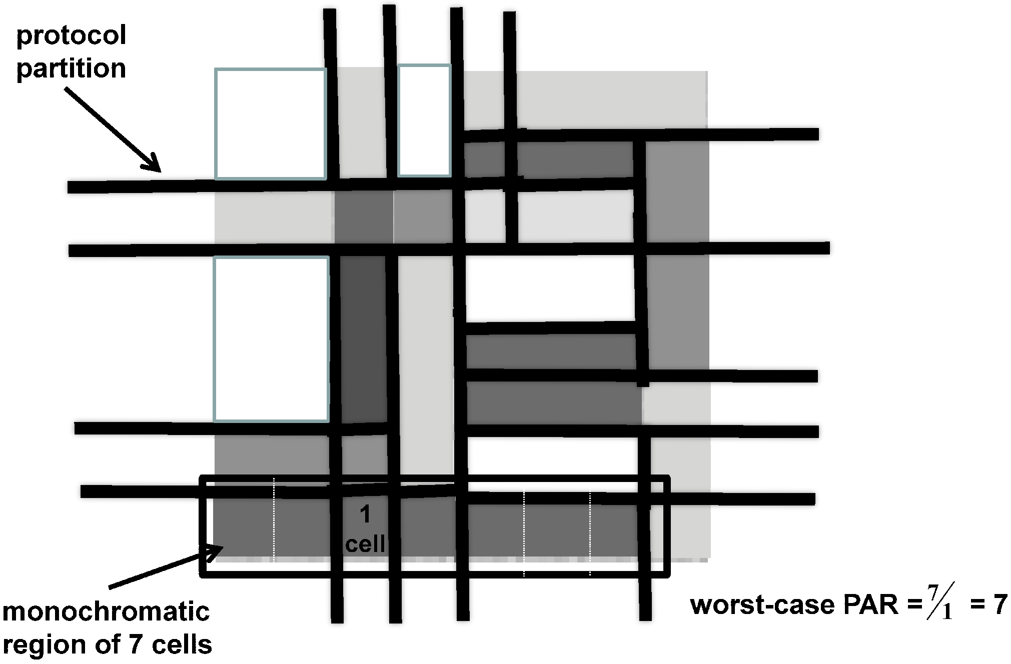


Fig. 10.7 Illustration of the calculation of the worst-case PAR Δ_{worst}.

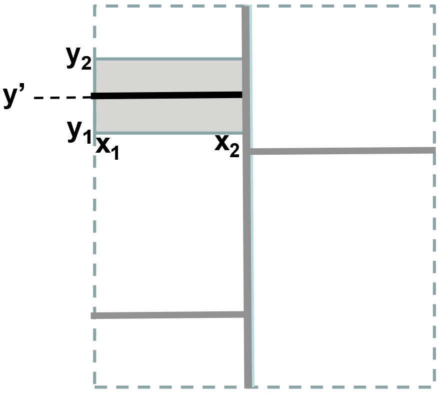


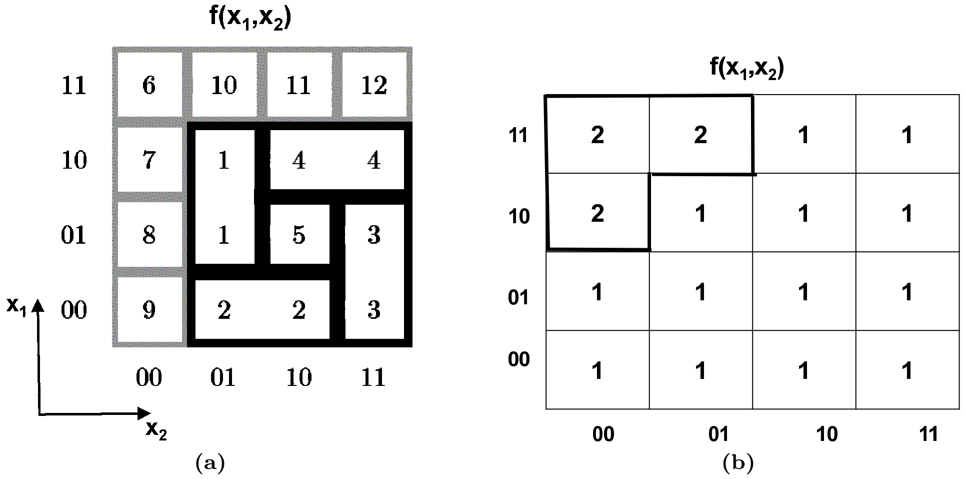
Fig. 10.8 Partitioning an ideal monochromatic rectangle leads to loss of privacy.

power of the more general dissection protocol in computing these tiling functions. The dissection protocol was defined in Section 10.4. To illustrate the main ideas more clearly, here we consider a slightly simplified version of the definition of a tiling function in which we assume that the index of a row (respectively, a column) of A_f is the same as the private value held by the first (respectively, second) agent. Then, a function f is a tiling function [Comi *et al.* (2012)] if the monochromatic

Table 10.1 List of bounds on average and worst-case Par for a few functions as derived in Feigenbaum *et al.* (2010). $\mathcal{D}$ is assumed to be the uniform distribution.

Protocol	Computed function		
	$f_{\text{millionaire}}$	$f_{\text{Second-priceVickrey}}$	
	$\Delta_{\mathcal{D}}$	Δ_{worst}	$\Delta_{\mathcal{D}}$
arbitrary	$\geq 2^k - \frac{1}{2} - \frac{1}{2^{k+1}}$		$\leq \frac{2}{3}2^k + \frac{1}{3\,2^k}$
bisection	$= \frac{3}{2}2^k - \frac{1}{2}$	$= 2^{\frac{k}{2}}$	$= \frac{k}{2} + 1$
α-bisection		$= 2^{\frac{k}{2}}$ (assuming $\alpha > \frac{1}{2^k}$)	
bounded-bisection$_{g(k)}$			$= \frac{g(k)+3}{2} - \frac{2^{g(k)}}{2^{k+1}} + \frac{1}{2^{k+1}} - \frac{1}{2^{g(k)+1}}$
sealed-bid auction			$= \frac{2^{k+1}}{3} + \frac{1}{3\,2^k}$
Ascending-price English auction			$= 1$

regions in A_f form a tiling; the number of monochromatic regions in this tiling is denote by ∇_f. See 10.9 for illustrations. Comi et al. [Comi *et al.* (2012)] proved the following results.



(a)

x_1 \ x_2	00	01	10	11
11	6	10	11	12
10	7	1	4	4
01	8	1	5	3
00	9	2	2	3

(b)

	00	01	10	11
11	2	2	1	1
10	2	1	1	1
01	1	1	1	1
00	1	1	1	1

Fig. 10.9 Example of **(a)** tiling and **(b)** non-tiling functions.

Theorem 10.2.

(a) Every boolean tiling function can be computed in a perfectly privacy-preserving manner.

(b) There exists a tiling function $f : \{0,1\}^k \times \{0,1\}^k \mapsto \{0,1\}^3$ such that every dissection protocol $\mathcal{P}$ for f has $\Delta_{\text{worst}} = \Omega\left(2^{\frac{k}{2}}\right)$.

(c) Let $\mathcal{D}$ be the uniform distribution. Then, the following results hold.

(i) *For any tiling function f, there is a dissection protocol $\mathcal{P}$ using at most* $\mathrm{O}\left(\nabla_f\right)$ *communication rounds such that*

- $\Delta_{\mathcal{D}} \leq 4$, *and*
- $\mathcal{P}$ *can be computed in* $\mathrm{O}\left(k4^k\right)$ *time.*

(ii) *There exists a tiling function f such that for every dissection protocol we have* $\Delta_{\mathcal{D}} \geq \frac{11}{9}$.

A proof of Theorem 10.2(c)(i) was obtained in Comi *et al.* (2012) via a connection between protocols and the binary space partitions (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, visibility problems, and motion planning [Tóth (2005)]. A Bsp for a collection of disjoint rectangles in the two-dimensional plane can be defined in the following manner. The plane is divided into two parts by cutting rectangles with a horizontal or vertical line if necessary. The two resulting parts of the plane are divided recursively in a similar manner and 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. Each leaf of the Bsp-tree then represents the final partitioning of the plane by storing at most one fragment of an input rectangle; see Fig. 10.10 for an illustration. The following result on Bsp was shown in d'Amore and Franciosa (1992):

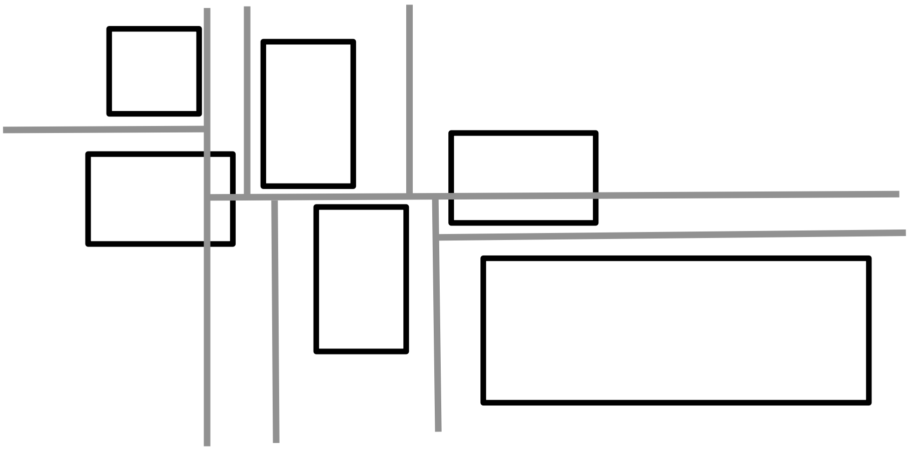


Fig. 10.10 A binary space partition for a set of given rectangles.

[d'Amore and Franciosa (1992)] For any set of disjoint axis-parallel rectangles in the plane, there is a Bsp such that every rectangle is partitioned into at most 4 rectangles due to Bsp.

The above result can be used to prove Theorem 10.2(c)(i) by identifying Bsps with dissection protocols.

10.5.2 *Generalization for $d > 2$ agents*

Comi et al. [Comi *et al.* (2012)] showed that the average Par is very high for dissection protocols even for 3 agents and uniform distribution, thereby suggesting that this quantification of privacy may not provide good bounds for three or more agents. More precisely, they proved the following result.

Theorem 10.3. *Let $\mathcal{D}$ denote the uniform distribution. Then, there exists a tiling function $f\colon \{0,1\}^k \times \{0,1\}^k \times \{0,1\}^k \mapsto \{0,1\}^{3k}$ such that every dissection protocol must have $\Delta_{\mathcal{D}} = \Omega\left(2^k\right)$.*

10.6 Conclusion

In this chapter, we have provided an overview to privacy preserving computing in a distributed function computation setup that includes game-theoretic settings. In particular, we have reviewed in greater details a recently developed geometric approach to quantifying loss of privacy. Future research questions of interest include identifying other non-tiling classes of functions for which good approximate-privacy preserving protocols are possible and relating the geometric privacy model to other privacy models.

References

Ada, A., Chattopadhyay, A., Cook, S., Fontes, L., Koucky, M. and Pitassi, T. (2012). The hardness of being private, in *Proc. IEEE Conference on Computational Complexity* (Porto, Portugal), pp. 192-202.

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

Ben-Or, M., Goldwasser, S. and Wigderson, A. (1988). Completeness theorems for non-cryptographic, fault-tolerant computation, in *Proc.* 20th *ACM Symposium on Theory of Computing* (Chicago, IL, USA), pp. 1-10.

Brandt, F. and Sandholm, T. (2008). On the existence of unconditionally privacy preserving auction protocols, *ACM Transactions on Information Systems Security* **11**, 2, pp. 1-21.

Chaum, D., Crépeau, C. and Damgaard, I. (1988). Multiparty, unconditionally secure protocols, in *Proc.* 20th *ACM Symposium on Theory of Computing* (Chicago, IL, USA), pp. 11-19.

Chor, B. and Kushilevitz, E. (1991). A zero-one law for boolean privacy, *SIAM Journal on Discrete Mathematics* **4**, pp. 36-47.

Comi, M., DasGupta, B., Schapira, M. and Srinivasan, V. (2012). On communication protocols that compute almost privately, *Theoretical Computer Science* **457**, pp. 45-58.

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

Dodis, Y., Halevi, S. and Rabin, T. (2000). A cryptographic solution to a game theoretic problem, in *Advances in Cryptology — CRYPTO 2000*, *Lecture Notes in Computer Science* **1880**, pp. 112-130.

Dwork, C. (2006). Differential privacy, in *Proc.* 33$^{\text{rd}}$ *International Colloquium on Automata, Languages and Programming* (Venice, Italy), pp. 1-12.

Feigenbaum, J., Jaggard, A. and Schapira, M. (2010). Approximate privacy: foundations and quantification, in *Proc.* 11$^{\text{th}}$ *ACM Conference on Electronic Commerce* (Cambridge, MA, USA), pp. 167-178.

Kushilevitz, E. (1992). Privacy and communication complexity, *SIAM Journal on Discrete Mathematics* **5**, 2, pp. 273-284.

Kushilevitz, E. and Nisan, N. (1997). Communication complexity (Cambridge University Press).

McGregor, A., Mironov, I., Pitassi, T., Reingold, O., Talwar, K. and Vadhan, S. (2010). The limits of two-party differential privacy, in *Proc.* 51$^{\text{st}}$ *IEEE Symposium on Foundations of Computer Science* (Las Vegas, NV, USA), pp. 81-90.

Naor, M., Pinkas, B. and Sumner, R. (1999). Privacy preserving auctions and mechanism design, in *Proc.* 1$^{\text{st}}$ *ACM Conference on Electronic Commerce* (Denver, CO, USA), pp. 129-139.

Nisan, N., Roughgarden, T., Tardos, E. and Vazirani, V. (2007). Algorithmic game theory (Cambridge University Press).

Sweeney, L. (2002). k-anonymity: a model for protecting privacy, *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems* **10**, 5, pp. 557-570.

Tóth, C. D. (2005). Binary space partitions: recent developments, in J. E. Goodman, J. Pach and E. Welzl (eds.), *Combinatorial and Computational Geometry* (MSRI Publications **52**, Cambridge University Press), pp. 529-556.

Vickrey, W. (1961). Counterspeculation, auctions and competitive sealed tenders, *Journal of Finance* **16**, 1, pp. 8-37.

Yao, A. C. (1979). Some complexity questions related to distributive computing, in *Proc.* 11$^{\text{th}}$ *ACM Symposium on Theory of Computing* (Atlanta, GA, USA), pp. 209-213.

Yao, A. C. (1982). Protocols for secure computation, in *Proc.* 23$^{\text{rd}}$ *IEEE Symposium on Foundations of Computer Science* (Chicago, IL, USA), pp. 160-164.

Yao, A. C. (1986). How to generate and exchange secrets, in *Proc.* 27$^{\text{th}}$ *IEEE Symposium on Foundations of Computer Science* (Toronto, Canada), pp. 162-167.