Privacy, Additional Information, and Communication *

Reuven Bar-Yehuda † Benny Chor ‡ Eyal Kushilevitz § Alon Orlitsky ¶

Abstract

Two parties, each holding one input of a two-variable function, communicate in order to determine the value of the function. Each party wants to expose as little of its input as possible to the other party. We prove tight bounds on the minimum amount of information about the individual inputs that must be revealed in the computation of most functions and of some specific ones, and show that a computation that reveals little information about the individual inputs may require many more message exchanges than a more revealing computation.

Key words: Private distributed protocols, additional-information, communication-complexity, rounds-complexity.

1 Introduction

Let f be a function of two n-bit inputs, x and y. Two honest parties, $P_{\mathcal{X}}$ holding x and $P_{\mathcal{Y}}$ having y, each with unlimited computing power, communicate to determine $f(x, y)$. Each party wants to keep as much of its input secret from the other party.

For some *privately computable* functions this can be done without revealing any more information about each input than is implied by the value of the function and the other input. For example, if $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n)$, and

$$f(x, y) \stackrel{\text{def}}{=} \sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i$$

*Research of B. Chor and E. Kushilevitz was supported in part by US-Israel BSF grant 88-00282. Part of B. Chor's research was done while visiting AT&T Bell Laboratories, Murray Hill, NJ. An early version of this paper appeared in *Proc. of 5th IEEE Structure in Complexity Theory*, 1990, pp. 55–65.

†Department of Computer Science, Technion, Haifa 32000, Israel e-mail reuven@cs.technion.ac.il

‡Department of Computer Science, Technion, Haifa 32000, Israel. e-mail benny@cs.technion.ac.il

§Department of Computer Science, Technion, Haifa 32000, Israel. e-mail eyalk@das.harvard.edu

¶AT&T Bell Laboratories, Murray Hill, New Jersey, USA. e-mail alon@research.att.com

is the total *Hamming weight* of x and y, then $P_{\mathcal{X}}$ can transmit the Hamming weight of x, while $P_{\mathcal{Y}}$ transmits the Hamming weight of y. Each party can now compute $f(x, y)$, and none has gained more information about the other party's input than is implied by $f(x, y)$ and the input it holds.

Ben-Or, Goldwasser, and Wigderson [3] initiated the study of privately-computable functions and presented a function that is not privately computable. Chor and Kushilevitz [4] characterized the privately computable Boolean functions. Subsequently, Kushilevitz [10] and Beaver [2] gave a complete characterization of the (general valued) privately-computable functions. It followed from these characterizations that "most" functions are not privately computable.

For functions that are not privately computable, it is natural to study the minimum amount of information about the individual inputs that must be divulged in their computation. There are several ways to quantify this *privacy cost* of a function and in this paper we consider three of them: a combinatorial measure $\mathcal{I}_{\text{c}}$, an information-theoretic measure $\mathcal{I}_{\text{i}}$, and a measure $\mathcal{I}_{\text{c-i}}$ that has both combinatorial and information-theoretic aspects.

For concreteness, we first define (Section 3) and analyze (Sections 4 and 5) $\mathcal{I}_{\text{c}}$. Following are some of the results obtained. Every Boolean function f over $\{0,1\}^n \times \{0,1\}^n$ can be computed while revealing at most $n+1$ information bits: $P_{\mathcal{X}}$ transmits x and $P_{\mathcal{Y}}$ responds by transmitting $f(x, y)$; therefore $\mathcal{I}_{\text{c}}(f) \leq n+1$. We show that for most such functions this bound is tight: $\mathcal{I}_{\text{c}}(f) \geq n - \log n - 3$. We prove linear lower bounds on $\mathcal{I}_{\text{c}}(f)$ for some explicit functions, including an $n/2 - 2$ lower bound for all functions whose function matrix is of *Hadamard type* (e.g., *inner product modulo 2*), and show that the privacy cost of the *identity* and *comparison* functions are about $\log n$.

Every function can be computed by exchanging just two messages. Kushilevitz [10] showed that certain functions require up to 2^{n+1} messages to be computed privately. One may ask whether compromising some privacy can help reduce the number of messages required. In Section 6 we show that, at the cost of revealing k extra bits of information, any function can be computed using $O(k \cdot 2^{(2n+1)/(k+1)})$ messages. Furthermore, for some functions, even a single extra bit of revealed information can decrease the number of messages required from 2^{n+1} to 2.

In Section 7 we define and investigate $\mathcal{I}_{\text{i}}$ and $\mathcal{I}_{\text{c-i}}$. Among other results, we prove that $\mathcal{I}_{\text{i}}(f) \leq \mathcal{I}_{\text{c-i}}(f) \leq \mathcal{I}_{\text{c}}(f)$ for every function f, and that for most Boolean functions over $\{0,1\}^n \times \{0,1\}^n$, including inner product modulo 2, $\mathcal{I}_{\text{c-i}}(f)$ and $\mathcal{I}_{\text{i}}(f)$ are about $n/2$.

The next section briefly introduces the communication model. Section 8 contains some concluding remarks.

2 Communication and Computation

We describe the communication model, delaying the cryptographic setup to the next section. Only those definitions and results essential to this work are presented. Additional details and proofs can be found in Yao [13].

As before, f is a function of two n-bit inputs x and y. Two parties, $P_{\mathcal{X}}$ holding x and $P_{\mathcal{Y}}$ having y, each with unlimited computing power, want to determine $f(x, y)$. To that end, $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ alternate in transmitting messages over a noiseless binary channel according to an agreed-upon communication protocol. We consider two kinds of protocols.

When *deterministic protocols* are used, each message is determined by the input known to its transmitting party and by previous messages. We require that in every round of communication, the set of all possible messages be prefix-free. This enables the receiver to know when the message ends. A protocol *computes f*, or is *a protocol for f*, if for every (x, y) both parties can determine $f(x, y)$ given their input and the exchanged messages.

The *communication string* $\mathcal{A}(x, y)$ of a protocol $\mathcal{A}$ on an input (x, y) is the concatenation of the messages sent in the course of the execution of $\mathcal{A}$ when $P_{\mathcal{X}}$ has x and $P_{\mathcal{Y}}$ has y. The prefix-freeness of the messages implies that the communication string can be uniquely decomposed into the individual messages.

A *rectangle* (in $\{0,1\}^n \times \{0,1\}^n$) is a Cartesian product $A \times B$ where $A, B \subseteq \{0,1\}^n$. The rectangle is f-*constant*, or *monochromatic* if f is constant over it. Let $\mathcal{A}$ be a deterministic protocol. Fore a string s, define

$$\mathcal{A}^{-1}(s) \stackrel{\text{def}}{=} \{(x, y) : A(x, y) = s\}$$

to be the *inverse image* of $\mathcal{A}$ when viewed as a function over $\{0,1\}^n \times \{0,1\}^n$. It can be shown that $\mathcal{A}^{-1}(s)$ is a rectangle and that if $\mathcal{A}$ computes f, then $\mathcal{A}^{-1}(s)$ is f-constant. We therefore say that a protocol $\mathcal{A}$ for f partitions $\{0,1\}^n \times \{0,1\}^n$ into f-constant rectangles.

The *complexity* of $\mathcal{A}$ is the maximum, over all inputs, of number of bits in their communication string. It is the largest number of bits the two parties may have to exchange when using the protocol. $C_D(f)$, the *deterministic communication complexity* of f, is the smallest complexity of a protocol for f. It is smallest number of bits that two parties computing f must exchange the worst case.

In *randomized protocols*, the two parties are allowed to randomize their transmissions and their decision as to the value of f. Still we require that in every round of communication, the set of all possible messages be prefix-free.

It is convenient (and, for our purposes, equivalent) to assume that prior to the commencement of communication each party makes a *random choice* that is independent of its (and the other party's) input. We can then assume that each message is determined by previous messages and by its transmitting party's input and initial random choice.

The *communication string* $\mathcal{A}(x, y)$ of a randomized protocol $\mathcal{A}$ on an input (x, y) is again the concatenation of all messages transmitted during an execution of $\mathcal{A}$ when $P_{\mathcal{X}}$ has x and $P_{\mathcal{Y}}$ has y. This time, however, $\mathcal{A}(x, y)$ is a random string.

At the end of communication, each party decides on the value of f, which again is a random variable. It is convenient to require that the two parties always agree on this random value which we denote by $L_{\mathcal{A}}(x,y)$. Since we are not concerned with the number of transmitted bits, this is easy to achieve — for example, $P_{\mathcal{X}}$ can transmit $L_{\mathcal{A}}(x,y)$ at the end of communication. We say that $\mathcal{A}$ *computes* f, or is a *protocol for* f, if for all $x, y \in \{0,1\}^n$,

$$Pr\left(L_{\mathcal{A}}(x,y) = f(x,y)\right) > \frac{1}{2} \ .$$

By taking expectations, one can define the complexity of f, very much like we did for deterministic protocols, but we will not use this measure.

3 Measuring Additional Information – The Combinatorial Measure $\mathcal{I}_{\mathbf{C}}$

We present two formalizations for the privacy cost of f — a weak definition and a stronger one. Other measures are introduced in section 7.

Let h, like f, be a function of two n-bit strings. Intuitively, we say that a protocol $\mathcal{A}$ for f *reveals at most* h, or is *h-private*, if all the information that a given party learns about the other input during the execution of $\mathcal{A}$ can be inferred from its own input and the value of $h(x,y)$[1].

Definition 1: *A protocol $\mathcal{A}$ for f is* weakly h–private for $P_{\mathcal{X}}$ *if for every $x, y_1, y_2 \in \{0,1\}^n$, $h(x,y_1) = h(x,y_2)$ implies that $\mathcal{A}(x,y_1)$ and $\mathcal{A}(x,y_2)$ have the same distribution, namely, for every communication string s,*

$$Pr\left(s = \mathcal{A}(x,y_1)\right) \ = \ Pr\left(s = \mathcal{A}(x,y_2)\right)$$

where the probability is taken over the random choices $r_{\mathcal{X}}$ and $r_{\mathcal{Y}}$.

In the stronger definition, we require that the protocol make no errors in computing $f(x,y)$ and that the communication on the two instances should be identically distributed given not only the input of each party (x or y), but also given its random choices ($r_{\mathcal{X}}$ or $r_{\mathcal{Y}}$). Formally,

Definition 2: *A protocol $\mathcal{A}$ for f is* strongly h–private *for $P_{\mathcal{X}}$ if*

- *For every $x, y \in \{0,1\}^n$:*
$$Pr(L_{\mathcal{A}}(x,y) = f(x,y)) \ = \ 1 \ .$$
- *For every $x, y_1, y_2 \in \{0,1\}^n$, $h(x,y_1) = h(x,y_2)$ implies that for all random choices $r_{\mathcal{X}}$ of $P_{\mathcal{X}}$, $\mathcal{A}(x,y_1)$ and $\mathcal{A}(x,y_2)$ have the same distribution, namely, for every communication string s,*
$$Pr\left(s = \mathcal{A}(x,y_1) | r_{\mathcal{X}}\right) = Pr\left(s = \mathcal{A}(x,y_2) | r_{\mathcal{X}}\right)$$
where the probability is taken over the random choices $r_{\mathcal{Y}}$ of $P_{\mathcal{Y}}$.

[1] Our definition is related to that of "an encryption function for f which leaks at most h", by Abadi, Feigenbaum, and Kilian [1]. However, the contexts as well as the formulations are different.

Weak and strong h–privacy for $P_{\mathcal{Y}}$ are defined symmetrically.

Definition 3: *Let h_1 and h_2 be functions of two n-bit inputs. A protocol $\mathcal{A}$ is* weakly (strongly) (h_1, h_2)–private *if it is weakly (strongly) h_1–private for $P_{\mathcal{X}}$ and weakly (strongly) h_2–private for $P_{\mathcal{Y}}$.*

It is convenient to identify h_1 and h_2 in the above definition.

Definition 4: *A protocol $\mathcal{A}$ is* weakly (strongly) h-private *if it is weakly (strongly) (h, h)-private.*

Definition 5: *A function f is* weakly (strongly) h-private *if it has a weakly (strongly) h-private protocol.*

Privately-computable functions [4, 10, 2] are a special case of the last definition:

Definition 6: *A function f is* weakly (strongly) private *if it is weakly (strongly) f-private.*

We now define the amount of revealed information required to compute f. The definition uses only the weak version of h-privacy. The reason we do not give the corresponding strong version will become apparent in section 4, when we show that weak and strong h-privacy are equivalent requirements.

An h-private protocol reveals at most the value of h during its execution. Intuitively, the amount of additional information released is related to $|\text{range}(h)|$. We quantify this amount as $|\text{range}(h)|$. (All logarithms are to the base 2.)

Definition 7: *The* revealed information *required for computing f is*

$$\mathcal{I}_c(f) \stackrel{def}{=} \min\{\log_2 |range(h)| : \ f \text{ is weakly } h\text{-private} \} \ .$$

The additional information *required for computing f, is*

$$\mathcal{E}(f) \stackrel{def}{=} \mathcal{I}_c(f) - \log_2 |range(f)| \ .$$

Note that a function is privately-computable if and only if $\mathcal{I}_C(f) = \log_2 |\text{range}(f)|$, i.e., if and only if $\mathcal{E}(f) = 0$.

Our first result is a trivial relation between the communication complexity of a function and its privacy cost.

Lemma 1: *For every f,*

$$\mathcal{I}_c(f) \leq C_D(f) \ .$$

Proof: *Let $\mathcal{A}$ be a deterministic protocol computing f and achieving $C_D(f)$. View $\mathcal{A}$ as a function over $\{0,1\}^n \times \{0,1\}^n$. Clearly, $\mathcal{A}$ is an $\mathcal{A}$-private protocol for f, and $\log |range(\mathcal{A})| \leq C_D(f)$.* □

As could be expected, this simple bound is not tight. Examples 1 and 2 (Section 5) show that there are functions whose communication complexity is $n + 1$ but their privacy cost is only $\log n$.

3.1 Related Results

We will use the following definitions and results. View the function f as the $2^n \times 2^n$ matrix M_f whose $(x,y)'th$ entry is $f(x,y)$. A submatrix of M_f is a restriction of M_f to a set of rows and a set of columns (not necessarily contiguous). A constant submatrix is called *monochromatic*.

Let M be a matrix. We say that two row indices x_1 and x_2 are *related* and write $x_1 \sim x_2$ if there is a coordinate y in which the two rows agree: $M_{x_1,y} = M_{x_2,y}$. This relation is clearly reflexive and symmetric, we define the relation $\equiv$ to be the transitive closure of $\sim$. In a similar way, we define the relations $\sim$ and $\equiv$ on the matrix columns. A matrix is *forbidden* if it is not monochromatic, all its rows are in the same equivalence class, and all its columns are in the same equivalence class (with respect to $\equiv$). The privately-computable functions can be characterized as follows:

Privacy-Characterization Theorem [10, 2] The following are equivalent: (1) f is weakly private, (2) f is strongly private, (3) M_f does not contain any forbidden submatrix.

If M_f does not contain a forbidden submatrix, it can be recursively decomposed, alternately by rows and by columns, into disjoint submatrices, such that no decomposition partitions an equivalence class, and the final submatrices are monochromatic. This decomposition corresponds to a tree of submatrices of M_f. The root of the tree is M_f, the children of every node are either a row- or a column-decomposition of the node, and the leaves are monochromatic matrices. Let $d(f)$ be the minimum depth of such a tree for M_f. We have:

Round-Complexity Theorem [10]: $d(f)$ is the minimum number of messages exchanged in a private computation of f. Furthermore, there is a deterministic protocol exchanging this number of messages.

Using this theorem, one can analyze the message-complexity of private functions and construct private functions over $\{0,1\}^n \times \{0,1\}^n$ requiring up to $2 \cdot (2^n - 1)$ messages in any private computation (the highest achievable value).

4 Relations to Private Computations

We prove the equivalence of weak and strong h-privacy. This equivalence is useful in proving lower bounds on $\mathcal{I}_{\mathrm{C}}(f)$, as it means that it suffices to consider *deterministic* protocols that compute f. Let $f_h(x,y)$ be the function f concatenated with h (i.e. $f(x,y) \circ h(x,y)$). We relate the question of whether f has an f_h-private protocol to the question of whether f_h is private. Since any computation of f reveals at least f's value, it is convenient to consider f_h-private protocols for f: The amount of additional information in such protocol is at most $\log_2 |range(h)|$.

Theorem 1: *Let f and h be arbitrary functions. If f_h is weakly private then f is strongly f_h-private.*

Proof: *Assume that f_h is weakly private. By the Privacy-Characterization Theorem, f_h is strongly private, and by the Round-Complexity Theorem, there exists a* deterministic *protocol $\mathcal{A}$ which computes f_h strongly privately. We modify $\mathcal{A}$ to a new protocol $\mathcal{A}'$ by adding one more message, in which the party receiving the last message in $\mathcal{A}$ (consisting of $f_h(x,y)$) sends $f(x,y)$. It is easy to verify that the modified protocol computes f and is strongly h–private.* □

The converse does not hold. An f_h-private protocol for f need not compute h, and therefore cannot in general be transformed into a private protocol for f_h. For example, a constant f is f_h-private for any h, but if h is not private then certainly f_h is not private.

However, it is possible to prove a modified version of the reverse direction: If f is f_h-private, then there is a function g, whose range is a subset of range$(h) \times$ range(h), such that f_g is private. We start with a lemma.

Lemma 2: *Let f, h_1 and h_2 be arbitrary functions. f is weakly (f_{h_1}, f_{h_2})-private if and only if f is strongly (f_{h_1}, f_{h_2})-private.*

Proof: Let $\mathcal{A}$ be a weakly (f_{h_1}, f_{h_2})-private protocol for f. Suppose that $P_{\mathcal{X}}$ sends the first message, m_1, in $\mathcal{A}$. Define an equivalence relation $\equiv_1$ on the input pairs (x,y) by $(x_1,y) \equiv_1 (x_2,y)$ if and only if the messages sent by $P_{\mathcal{X}}$ in on inputs x_1 and x_2 are identically distributed. Similarly, $\equiv_2$ is defined according to the distribution of m_2 (given the previous message m_1), and it refines the equivalence classes of $\equiv_1$. In general, the equivalence relation $\equiv_i$ is defined according to the distribution of the i-th message m_i (given all previous messages). The relation $\equiv_i$ decomposes the input pairs (x,y) into equivalence classes $C_1^i, C_2^i, \ldots, C_t^i$, refining the $\equiv_{i-1}$ equivalence classes in a manner depending only on the input (x or y) of the corresponding processor ($P_{\mathcal{X}}$ or $P_{\mathcal{Y}}$). As there are only 2^{2n} input pairs, the number of equivalence classes, t, cannot exceed 2^{2n}. Furthermore, the number of messages i required till $\equiv_i$ stabilizes is finite, even if $\mathcal{A}$ has infinite runs.

These observations lead to the modified protocol, $\mathcal{A}'$. In the i-th message of $\mathcal{A}'$, the corresponding party sends the index j of the $\equiv_i$ equivalence class, as determined by its input (x or y) and the $\equiv_{i-1}$ equivalence class. The protocol terminates when the current equivalence class is stable (namely it does not get more refined in future messages). At this point both parties know *exactly* the distribution of messages $\mathcal{A}(x,y)$ on the input pair (x,y). Since the probability that the last message sent in $\mathcal{A}$ equals $f(x,y)$ is strictly greater than $1/2$, the parties can determine $f(x,y)$ (with no errors) and send it as the last message. Thus $\mathcal{A}'$ is a deterministic protocol for computing f.

We now show that $\mathcal{A}'$ is (f_{h_1}, f_{h_2})-private. Let (x,y_1) and (x,y_2) satisfy $f(x,y_1) = f(x,y_2)$ and $h_1(x,y_1) = h_1(x,y_2)$. Since $\mathcal{A}$ is weakly (f_{h_1}, f_{h_2})-private, the distributions of messages $\mathcal{A}(x,y_1)$ and $\mathcal{A}(x,y_2)$ are identical. This implies that in $\mathcal{A}'$ the same communication string is sent on both input-pairs (x,y_1) and (x,y_2) and so $\mathcal{A}'$ is f_{h_1}-private for $P_{\mathcal{X}}$. Similar argument holds for $P_{\mathcal{Y}}$, and therefore $\mathcal{A}'$ is (f_{h_1}, f_{h_2})-private. As $\mathcal{A}'$ is a deterministic protocol, it is necessarily strongly (f_{h_1}, f_{h_2})-private. □

Theorem 2: *Let f, h_1 and h_2 be arbitrary functions. If f is weakly (f_{h_1}, f_{h_2})-private then there exists a pair of functions g_1 and g_2 with $|range(g_i)| \leq |range(h_i)|$ $(i = 1, 2)$ such that $f_{g_1 \circ g_2}$ is strongly private.*

Proof: According to Lemma 2, there is a deterministic (f_{h_1}, f_{h_2})-private protocol $\mathcal{A}$ which computes f. In Section 2, we defined for every communication string s, $\mathcal{A}^{-1}(s) = \{(x, y) | \mathcal{A}(x, y) = s\}$ and noted that the set $\mathcal{A}^{-1}(s)$ is a generalized rectangle $\mathcal{A}^{-1}(s) = C(s) \times D(s)$ $(C(s), D(s) \subseteq \{0,1\}^n)$. Define the function g_1 such that for every communication string s, the value of $g_1(x, y)$ is constant over every row in $C(s)$. That is, for every $x \in C(s)$

$$g_1(x, y) = h_1(x, y_s)$$

where y_s is some fixed arbitrary element of $D(s)$. We define $g_2(x, y)$ similarly, such that $g_2(x, y)$ is constant over every column in $D(s)$.

Clearly range$(g_i) \subseteq$ range(h_i) $(i = 1, 2)$. For the function g_1, the party $P_{\mathcal{X}}$ can determine $g_1(x, y)$ given x and $\mathcal{A}(x, y)$. Similarly $P_{\mathcal{Y}}$ can determine $g_2(x, y)$ given y and $\mathcal{A}(x, y)$. This property makes it possible to transform the protocol $\mathcal{A}$ into a new protocol $\mathcal{A}'$ which computes $f_{g_1 \circ g_2}$ privately.

We first show that $\mathcal{A}$ is a (g_1, g_2)-private protocol for computing f. Suppose $f(x, y_1) = f(x, y_2)$ and $g_1(x, y_1) = g_1(x, y_2)$. We show that this implies $\mathcal{A}(x, y_1) = \mathcal{A}(x, y_2)$. In other words we should prove that (x, y_1) and (x, y_2) are in the same $\mathcal{A}^{-1}(s)$ (similar proof holds for pairs of the type (x_1, y) and (x_2, y)). Suppose towards a contradiction that $(x, y_1) \in \mathcal{A}^{-1}(s_1)$ and $(x, y_2) \in \mathcal{A}^{-1}(s_2)$. Since $g_1(x, y_1) = g_1(x, y_2)$ then there are $y_1' \in D(s_1)$ and $y_2' \in D(s_2)$ such that $h_1(x, y_1') = h_1(x, y_2')$. But then the communication on (x, y_1) and (x, y_2) differ - contradicting the fact that $\mathcal{A}$ is (f_{h_1}, f_{h_2})-private.

We modify the protocol $\mathcal{A}$ and get a new protocol $\mathcal{A}'$ which computes $f_{g_1 \circ g_2}$. The protocol $\mathcal{A}'$ is similar to $\mathcal{A}$ with two additional messages in which $P_{\mathcal{X}}$ computes $g_1(x, y)$ (note that x and $\mathcal{A}(x, y)$ completely determines the value of $g_1(x, y)$) and $P_{\mathcal{Y}}$ computes $g_2(x, y)$. Since $\mathcal{A}$ computes f (f_{g_1}, f_{g_2})-privately, it is easy to verify that $\mathcal{A}'$ privately computes $f_{g_1 \circ g_2}$. □

Corollary 1: *If f is weakly f_h–private then there is a function g with $|range(g)| \leq |range(h)|^2$, such that f_g is strongly private.*

5 Bounds on $\mathcal{I}_{\mathbf{C}}$

We bound the privacy cost of various random and explicit functions. Upper bounds are proved by demonstrating the appropriate protocols and lower-bound proofs employ communication-complexity techniques in conjunction with Lemma 2. We begin with three examples.

Example 1*: View x and y as integers in $\{0, \ldots, 2^n - 1\}$ and consider the function*

$$\min(x,y) \stackrel{def}{=} \begin{cases} x & \text{if } x \le y, \\ y & \text{otherwise.} \end{cases}$$

We show that min *is not private and that* $0 < \mathcal{E}(\min) \le 1$.

For every a, $\min(a,a) = \min(a,a+1) = \min(a+1,a) = a$ *and* $\min(a+1,a+1) = a+1$. *Hence* $M_{\min}$ *contains a forbidden submatrix (see Subsection 3.1 for definition). The Privacy-Characterization Theorem implies that* min *is not private and so* $\mathcal{E}(\min) > 0$.

To show that $\mathcal{E}(\min) < 1$, *consider the following protocol.*

For i *from* 0 *to* $2^n - 1$, *or until the first '1' is transmitted:*

(1) $P_{\mathcal{X}}$ *transmits '1' if* $x = i$ *and transmits '0' otherwise,*

(2) $P_{\mathcal{Y}}$ *transmits '1' if* $y = i$ *and transmits '0' otherwise.*

It is easy to verify that when communication ends, $P_{\mathcal{X}}$ *knows only* $\min(x,y)$ *(if* $x \le y$*, the protocol ends after step (1) for* $i = x$*) and that* $P_{\mathcal{Y}}$ *knows only* $\min(x,y)$ *and whether* $x = y$. *Let*

$$equ(x,y) \stackrel{def}{=} \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Then the protocol computes min *and is* $(\min \circ equ)$*-private. Hence,*

$$\mathcal{I}_c(\min) \le \log|range(\min \circ equ)| = \log|range(\min)| + 1 \ ,$$

and $\mathcal{E}(\min) = \mathcal{I}_c(\min) - \log|range(\min)| \le 1$. □

Example 2*: We show that for the equality function defined in Equation (1),*

$$\log n - 1 < \mathcal{I}_c(equ) \le \log(n+1) \ .$$

First we establish the upper bound. Let $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n)$, *and define*

$$h(x,y) \stackrel{def}{=} \begin{cases} \min\{i \mid x_i \neq y_i\} & \text{if } x \neq y, \\ n+1 & \text{if } x = y. \end{cases}$$

Consider the following protocol:

For i *from 1 to* n*:*

(1) $P_{\mathcal{X}}$ *sends* x_i *to* $P_{\mathcal{Y}}$.

(2) If $x_i \neq y_i$ *then* $P_{\mathcal{Y}}$ *transmits '0' and the execution terminates. If* $x_i = y_i$ *then* $P_{\mathcal{Y}}$ *sends '1'.*

It is easy to verify that the protocol computes equ correctly and that it is h-private as the only information revealed, besides the value of $f(x,y)$, is the first index i where $x_i \neq y_i$. Hence,

$$\mathcal{I}_C(f) \leq \log|range(h)| = \log(n+1)\ .$$

To prove the lower bound, let I_m be the $m \times m$ identity matrix. Note that I_m is the restriction of M_{equ} to the first m rows and columns. We prove by induction on m that any deterministic protocol $\mathcal{A}$ which computes the identity function on I_m, there is a $z \in \{0,1,\ldots,m-1\}$ such that the size of the set $R_z \stackrel{def}{=} \{\mathcal{A}(x,z) \,|\, 0 \leq x \leq m-1\} \cup \{\mathcal{A}(z,y) \,|\, 0 \leq y \leq m-1\}$ is at least $\log_2 m + 2$. We then show how this bound implies that if equ is h-private, then $|range(h)| \geq 1 + n/2$.

For $m = 2$, one can verify by inspection that for both values of z, $|R_z| = 3 = \log_2 2 + 2$. For $m > 2$, suppose, without loss of generality, that $P_{\mathcal{X}}$ sends in the first message one of j possible strings ($j \geq 2$). This first message partitions I_m into j "horizontal" rectangles. At least one of these rectangles contains at least $\lceil m/j \rceil$ diagonal points (z,z). By the structure of I_m, this rectangle contains a submatrix $F \times F$ which is isomorphic to $I(\lceil m/j \rceil)$. By the induction hypothesis, for at least one of the $z \in F$, the size of the induced set R_z (in $F \times F$) is at least $2 + \log_2\lceil m/j \rceil \geq 2 + \log_2 m - \log_2 j$. In addition, the set R_z contain at least $j-1$ communication strings from $I_m \backslash F \times F$ (as at least the first message sent on inputs in the other $j-1$ "horizontal" rectangles is different). Overall, $|R_z| \geq 2 + j - 1 - \log_2 j + \log_2 m \geq 2 + \log_2 m$ (with equality only for $j = 2$).

As $R_z = \{\mathcal{A}(x,z) \,|\, 0 \leq x \leq m-1\} \cup \{\mathcal{A}(z,y) \,|\, 0 \leq y \leq m-1\}$, the last inequality implies, without loss of generality, that the size of the set $\{\mathcal{A}(z,y) \,|\, 0 \leq y \leq m-1\}$ is at least $(2 + \log_2 m)/2$. Now if the protocol $\mathcal{A}$ is h-private, every y_1, y_2 for which $\mathcal{A}(z,y_1) \neq \mathcal{A}(z,y_2)$, must satisfy $h(z,y_1) \neq h(z,y_2)$. Therefore if equ in the domain I_m is h-private, then $|range(h)| \geq (2 + \log_2 m)/2$. Substituting $m = 2^n$, we have $|range(h)| \geq 1 + n/2$, and so $\mathcal{I}_C(f) > \log_2 n - 1$. □

Note [13] that $C_D(\text{equ}) \geq n$, hence $\mathcal{I}_C(\text{equ})$ is much smaller than $C_D(\text{equ})$.

Example 3*: A similar proof shows that the* comparison *function defined by:*

$$comp(x,y) \stackrel{def}{=} \begin{cases} 1 & \text{if } x \geq y, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

has

$$\log n - 1 < \mathcal{I}_C(comp) \leq \log(n+1)\ .$$

□

As mentioned in Section 2, every protocol for f partitions $\{0,1\}^n \times \{0,1\}^n$ into f-constant rectangles. The *width* of a rectangle $R = A \times B$ is $\min\{|A|,|B|\}$. R is *vertical* if $|A| \geq |B|$ and *horizontal* otherwise. Let r_f be the largest width of an f-constant rectangle. The next lemma relates r_f to $\mathcal{I}_C(f)$ and is useful in proving lower bounds on $\mathcal{I}_C(f)$ for certain functions.

Theorem 3: *For every function f defined over $\{0,1\}^n \times \{0,1\}^n$,*

$$\mathcal{I}_C(f) \geq n - \log r_f - 1 \ .$$

Proof: By Lemma 2 we need to consider only deterministic protocols. We show that if some deterministic protocol for f is h-private, then there is a row or a column over which h attains at least $2^{n-1}/r_f$ values.

Consider the rectangular partition of $\{0,1\}^n \times \{0,1\}^n$ inducted by the protocol. Without loss of generality, at least 2^{2n-1} elements of $\{0,1\}^n \times \{0,1\}^n$ belong to vertical rectangles (otherwise, interchange rows and columns below). Therefore, there is a row which has at least 2^{n-1} elements contained in vertical rectangles. As the protocol computes f, all the rectangles are f-constant, and the intersection of any row with a vertical f-constant rectangle has size of at most r_f. Hence, there must be at least $2^{n-1}/r_f$ rectangles intersecting that row.

Since the protocol is h-private, in any row (or column) h cannot obtain the same value over two different rectangles. Hence h obtains at least $2^{n-1}/r_f$ values over some row. $\square$

We use the lemma to lower bound $\mathcal{I}_C(f)$ for "most" and for some specific boolean functions. Let $\mathcal{F}_n$ be the set of Boolean functions over $\{0,1\}^n \times \{0,1\}^n$. Define

$$\mathcal{F}_n^* \stackrel{\text{def}}{=} \{f \in \mathcal{F}_n : r_f < 2n\}$$

to be the set of functions in $\mathcal{F}_n$ whose monochromatic squares all have width $< 2n$. The following standard argument shows that most functions in $\mathcal{F}_n$ have $r_f < 2n$.

Lemma 3:

$$|\mathcal{F}_n^*| \geq \left(1 - \frac{1}{[(2n)!]^2}\right) |\mathcal{F}_n| \ .$$

Proof: *Choose a function $f \in \mathcal{F}_n$ uniformly at random. If $r_f \geq 2n$, then M_f contains a monochromatic $2n$ by $2n$ subsquare. This happens with probability of at most:*

$$2 \cdot \frac{\binom{2^n}{2n}^2}{2^{(2n)^2}} < 2 \cdot \frac{\left[(2^n)^{2n}\right]^2}{[(2n)!]^2 \cdot 2^{(2n)^2}} = \frac{2}{[(2n)!]^2} \ .$$

$\square$

Combining the last lemma and Theorem 3, we obtain

Corollary 2: *A fraction of at least $1 - \frac{1}{[(2n)!]^2}$ of the Boolean functions $f \in \mathcal{F}_n$ have*

$$\mathcal{I}_C(f) > n - \log n - 2 \ .$$

$\square$

A ± 1 matrix is of *Hadamard type* [6] if every pair of rows and every pair of columns of the matrix are orthogonal. The function matrices of many interesting functions are of Hadamard type [12, pp. 45-48]. The next theorem applies to all of them.

Corollary 3: *Let $f \in \mathcal{F}_n$. If M_f is of Hadamard type, then*

$$\mathcal{I}_C(f) \geq \frac{n}{2} - 1 \ .$$

Proof: *A well known result by Lindsey [6, page 88] states all monochromatic rectangles in a 2^n by 2^n Hadamard matrix have width of at most $2^{n/2}$. Hence $r_f \leq 2^{n/2}$ and the theorem follows from Theorem 3.* $\square$

Example 4: *The n-variable inner-product-mod-2 function is defined by:*

$$ip(x, y) \stackrel{def}{=} \sum_{i=1}^{n} x_i \cdot y_i \pmod 2.$$

It is easy to verify that M_{ip} is a Hadamard matrix, hence

$$\mathcal{I}_C(ip) \geq \frac{n}{2} - 1 \ .$$

$\square$

6 Additional Information Versus Rounds Complexity

Every function f defined on $\{0,1\}^n \times \{0,1\}^n$ can be computed in two messages using $n+1$ bits. Kushilevitz [10] showed that for any $1 \leq m \leq 2 \cdot (2^n - 1)$, there are functions whose private computation requires m messages.

In this section, we explore a possible tradeoff between privacy and communication messages. We begin with two examples of private functions $f : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^{2n}$, whose message-complexity can be significantly reduced if additional information can be revealed. We then show a general tradeoff: If f is defined over $\{0,1\}^n \times \{0,1\}^n$ and $\mathcal{I}_C(f) \leq \ell$, then for every $0 \leq k \leq n - \ell$ there exists an $O(k \cdot 2^{2n/(k+1)})$-message protocol for f which reveals $\ell + k$ bits. This is meaningful when any computation of f that reveals at most ℓ information bits requires exponentially many messages (as we mentioned above, [10] showed that such functions exist).

Example 5: *Define g as follows (see Figure 1):*

$$g(x, y) = \begin{cases} -1 & \text{if } x = y \\ \min(x, y) & \text{if } x \neq y \end{cases}$$

f	0	1	2	3	$\cdots$	2^n-2	2^n-1
0	-1	0	0	0	$\cdots$	0	0
1	0	-1	1	1	$\cdots$	1	1
2	0	1	-1	2	$\cdots$	2	2
3	0	1	2	-1	$\cdots$	3	3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
2^n-2	0	1	2	3	$\cdots$	-1	2^n-2
2^n-1	0	1	2	3	$\cdots$	2^n-2	-1

Figure 1: The function g of Example 5

By [10], this function is private and has message-complexity of $2 \cdot (2^n - 1)$*. One simple way of trading additional information for message-complexity in* g *is the following:* $P_{\mathcal{X}}$ *starts the protocol by revealing the* k *most significant bits of its input,* x*, to* $P_{\mathcal{Y}}$*. If the* k *most significant bits of* y *are different from the* k *most significant bits of* x *then* $P_{\mathcal{Y}}$ *knows who holds the minimum. Otherwise, if the* k *most significant bits of both parties are the same, the domain of the function is reduced to* $n-k$ *bits, so that now* $2 \cdot (2^{n-k} - 1) + 1$ *messages of communication suffice.*

One can do even better – for every $1 \leq k \leq n$*, there is a protocol that reveals* $\log(k+1)$ *bits of additional information and takes* $2 \cdot (2^{n-k} + k - 1)$ *messages. For every* k *in the interval above, we define a function* h_k *as follows:*

$$h_k(x, y) = \max\{j \mid 0 \leq j \leq k \ , \quad x_1 \ldots x_j = y_1 \ldots y_j\}$$

(the length of common prefix of most significant bits in x *and* y*, up to* k*).*
The protocol goes as follows: the parties compute h_k *by exchanging bit by bit (at most* $2k$ *messages) until finding the first different bit. If* $h_k \neq k$ *then the parties know which of them holds the minimum and can compute* $g(x, y)$*. On the other hand, if* $h_k = k$ *then the parties execute the original protocol for* g *on the truncated inputs* $(x_{k+1} \ldots x_n, y_{k+1} \ldots y_n)$*. This takes at most* $2 \cdot (2^{n-k} - 1)$ *more messages.*

The next example demonstrates a function f with the strongest possible tradeoff. Any private computation f has message-complexity $2 \cdot (2^n - 1)$. However, while revealing only one bit of additional information, f can be computed in two messages (and using $O(n)$ information bits).

Example 6*:* *Define* f *as follows (see Figure 2):*

(1) $f(x, x) = f(x, x+1) = 2 \cdot x$ *(for* $x = 0, 1, \ldots, 2^n - 2$*).*

(2) $f(y+1, y) = f(y+2, y) = 2y + 1$ *(for* $y = 0, 1, \ldots, 2^n - 3$*).*

f	0	1	2	3	$\cdots$	2^n-2	2^n-1
0	0	0					
1	1	2	2				
2	1	3	4	4			
3		3	5	6	$\ddots$		
$\vdots$			$\ddots$	$\ddots$	$\ddots$	$\ddots$	
2^n-2					$\ddots$	$2^{n+1}-4$	$2^{n+1}-4$
2^n-1						$2^{n+1}-3$	$2^{n+1}-2$

Figure 2: The function f of Example 6

(3) Otherwise $f(x,y) = x \circ y$.

Take the Boolean function $h(x,y) = 1 \iff x = y$. Then the simple 2-message protocol in which $P_\mathcal{Y}$ sends y and $P_\mathcal{X}$ answers with $f(x,y)$ is f_h-private. Intuitively, the only privacy requirements with respect to $P_\mathcal{X}$ are given by the pairs of inputs of the form (x,x) and $(x,x+1)$. The function h gets different values on (x,x) and $(x,x+1)$ and thus in a f_h-private protocol $P_\mathcal{X}$ is allowed to distinguish between them.

The next theorem concerns the general tradeoff between additional information and the number of communication messages. Before stating it, we recall that private computations of private functions on the domain $\{0,1\}^n \times \{0,1\}^n$ might take up to $\Theta(2^n)$ communication messages [10].

Theorem 4: *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{2n}$ be a private function, and let $1 \leq k \leq n$. There is a protocol which computes f, reveals at most k bits of additional information, and requires $O(k \cdot 2^{(2n+1)/(k+1)})$ messages.*

Theorem 4 is meaningful when privately computing f requires "many" messages. It is extended to non-private functions in the following statement.

Theorem 5: *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{2n}$ be any function with $\mathcal{I}_C(f) \leq \ell$, and let $1 \leq k \leq n - \ell$. There is a protocol which computes f, reveals at most $\ell + k$ bits of information, and requires $O(k \cdot 2^{(2n+1)/(k+1)})$ messages.*

Proof of Theorem 5

By the supposition there is a function $g : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^\ell$, and a g-private protocol F for computing f. Using Lemma 2, we can assume that F is deterministic. Denote by T the

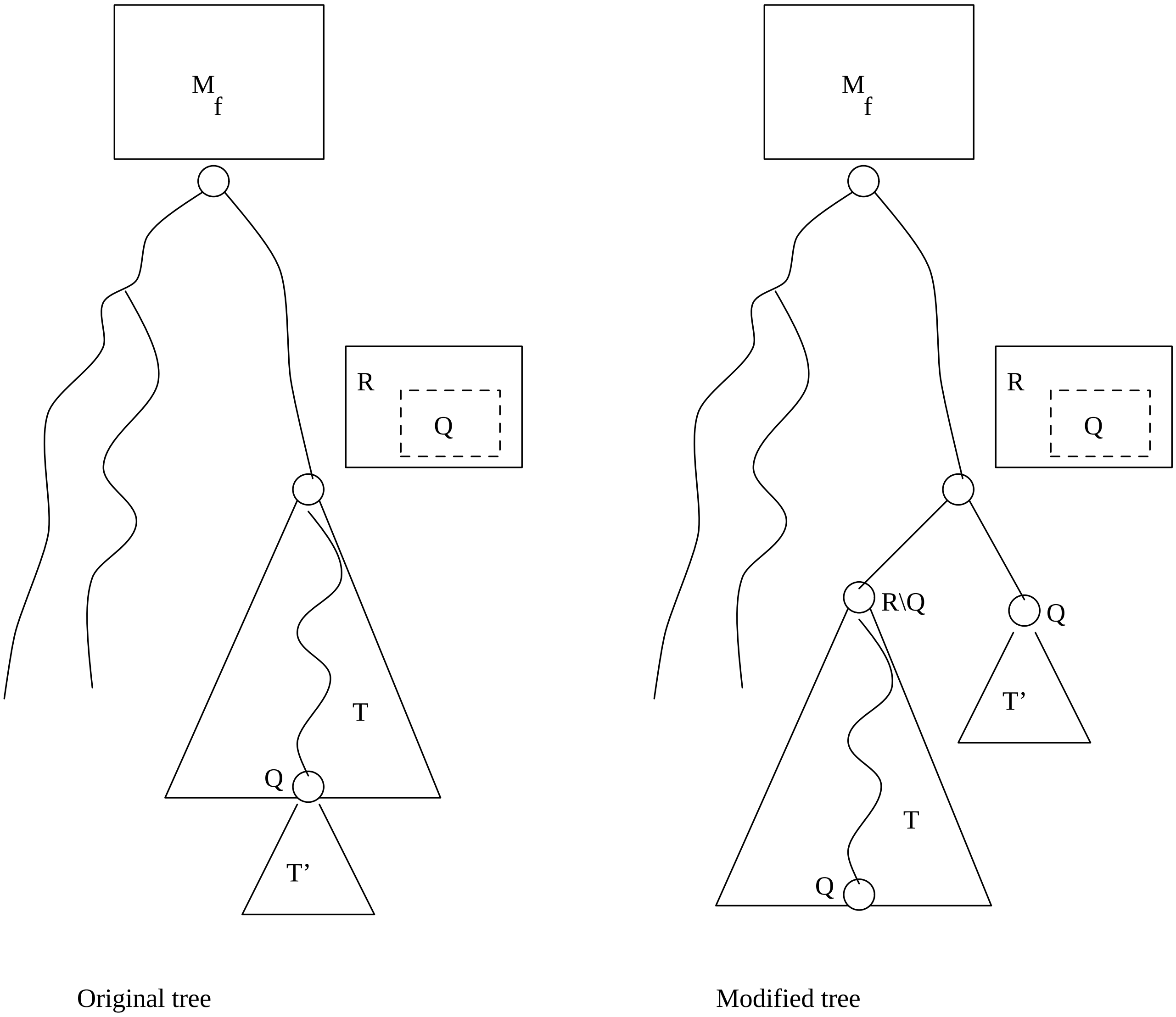

Figure 3: $ShortCut(R, Q)$

decomposition tree corresponding to the protocol F. Let M denote the number of nodes in T. Since the number of inputs (x, y) is 2^{2n}, and each leaf contains one or more inputs, $M \leq 2^{2n+1}$ (this assumes that every step of F makes some progress). We construct from F a protocol G which reveals at most $\ell + k$ bits of information, and uses at most $O(k \cdot M^{\frac{1}{k+1}})$ messages.

The main idea in the new protocol is to make appropriate "shortcuts" in the original tree. Suppose that the submatrices $R = C_1 \times D_1$ and $Q = C_2 \times D_2$ correspond to two nodes in the decomposition tree T, such that $Q \subseteq R$. This means that R and Q are on the same branch in T, with Q below R.

Modifying the original protocol by the $ShortCut(R, Q)$ operation causes the following change (see Figure 3): If the parties reach the R node, they first check if their input (x, y) is in Q. If it is, they make a "shortcut" to node Q, that is, they continue the protocol from that node. Otherwise

$((x,y) \in R \setminus Q)$, they continue the protocol from R as before. Formally, in the modified protocol, upon reaching $R = C_1 \times D_1$,

$$ShortCut(R,Q) = \begin{cases} \bullet & P_{\mathcal{X}} \text{ sends one bit indicating whether } x \in C_2. \\ \bullet & P_{\mathcal{Y}} \text{ sends one bit indicating whether } y \in D_2. \\ \bullet & \text{If } x \in C_2 \text{ and } y \in D_2 \text{ then the protocol continues from} \\ & Q = C_2 \times D_2. \text{ Otherwise, the protocol continues from the point } R. \end{cases}$$

Let d be the length of the path from node R to node Q in the original tree T. That is, for inputs $(x,y) \in Q$, d messages of communication are needed, after the parties find out that $(x,y) \in R$, to find out that $(x,y) \in Q$. In the new protocol, the number of messages needed after the parties find out that $(x,y) \in R$, to find out that $(x,y) \in Q$ is just 2, so the total number of messages (for $(x,y) \in Q$) is decreased by $d-2$. For $(x,y) \in (R \setminus Q)$, on the other hand, the number of messages in the modified protocol *increases* by 2, while for $(x,y) \notin R$, this number remains unchanged. In addition to the change in messages, the modified protocol might also reveal additional information. We claim that at most one additional bit is revealed by $ShortCut(R,Q)$: If (x,y) is in $R \setminus Q$ then $P_{\mathcal{Y}}$ learns whether $x \in C_2$, and $P_{\mathcal{X}}$ learns whether $y \in D_2$. For any other inputs (that is, $(x,y) \notin R$ or $(x,y) \in Q$) no extra information is given. Formally,

Lemma 4: *Let F be any g-private protocol for computing f. Let G be the protocol obtained from F by applying $ShortCut(R,Q)$. Let h be any Boolean function such that:*

$$h(x,y) = \begin{cases} 1 & \textit{if } (x,y) \in (C_1 \setminus C_2) \times (D_1 \setminus D_2) \\ 0 & \textit{if } (x,y) \in (C_1 \setminus C_2) \times D_2 \quad \textit{or} \quad (x,y) \in C_2 \times (D_1 \setminus D_2) \\ * & \textit{if } (x,y) \notin R \quad \textit{or} \quad (x,y) \in Q \end{cases}$$

where $*$ is a dontcare *(which can get arbitrary Boolean values for every (x,y) in the domain $Q \cup \bar{R}$). Then G is a $g \circ h$ – private protocol for computing f.*

Proof: Let (x,y_1) and (x,y_2) be two inputs with $g(x,y_1) = g(x,y_2)$, and $h(x,y_1) = h(x,y_2)$. (Similar proof holds for pairs of inputs of the form (x_1,y) and (x_2,y)). Since F is g-private and deterministic, the communication S_F satisfies $S_F(x,y_1) = S_F(x,y_2)$. Furthermore, the g-privacy of F implies that $(x,y_1) \in R \Leftrightarrow (x,y_2) \in R$ (otherwise, $P_{\mathcal{X}}$ can distinguish between (x,y_1) and (x,y_2) by the execution of F). Similarly $(x,y_1) \in Q \Leftrightarrow (x,y_2) \in Q$. Therefore we need consider only the three following cases:

- Both (x,y_1) and (x,y_2) are not in R. For such inputs the protocol G proceeds exactly the same as F, and thus $S_G(x,y_1) = S_G(x,y_2)$.

- Both (x,y_1) and (x,y_2) are in Q. For such inputs the protocol G works the same as F except that two messages are added, when reaching R, in which $P_{\mathcal{X}}$ sends one bit indicating that $x \in C_2$, and $P_{\mathcal{Y}}$ sends one bit indicating that $y_i \in D_2$ $(i = 1,2)$. Thus, $S_G(x,y_1) = S_G(x,y_2)$.

- Both (x, y_1) and (x, y_2) are in $R \backslash Q$. Since $h(x, y_1) = h(x, y_2)$ then either both $y_1, y_2 \in (D_1 \backslash D_2)$ or both $y_1, y_2 \in D_2$. Without loss of generality assume that $y_1, y_2 \in (D_1 \setminus D_2)$. For such inputs the protocol G works the same as F except that two messages are added, when reaching R, in which $P_{\mathcal{X}}$ sends one bit indicating that $x \in C_2$, and $P_{\mathcal{Y}}$ sends one bit indicating that $y_i \in (D_1 \setminus D_2)$ $(i = 1, 2)$. Thus, $S_G(x, y_1) = S_G(x, y_2)$.

Notice that our argument makes use of h's value only for $(x, y) \in R \setminus Q$. Therefore, h can get arbitrary values ($*$'s) outside $R \setminus Q$. □

A single shortcut can reduce the length of a single long branch, at the cost of revealing one extra bit of information. In order to reduce the length of many long branches, we would like to do *many* shortcuts simultaneously, and still reveal only a single bit of additional information. The key observation to this step is that if R_1, Q_1, R_2 and Q_2 correspond to nodes in F's decomposition tree with $Q_1 \subseteq R_1$, $Q_2 \subseteq R_2$, and $(R_1 \backslash Q_1) \cap (R_2 \backslash Q_2) = \emptyset$, then $ShortCut(R_1, Q_1)$ and $ShortCut(R_2, Q_2)$ can be done simultaneously. The functions h_1, h_2 corresponding to the two shortcuts have enough degrees of freedom which prevent conflicts in the relevant regions. This is since h_1 has non-$*$ values only inside $R_1 \setminus Q_1$, while h_2 has non-$*$ values only inside $R_2 \setminus Q_2$, and these two sets are disjoint. This argument clearly generalizes to any number of pairs R_i, Q_i with pairwise disjoint $R_i \setminus Q_i$. We now describe how to choose these pairs so that a significant length reduction is achieved.

Let V be a set of pairs of submatrices, (R_i, Q_i), corresponding to nodes on F's decomposition tree which satisfy $Q_i \subseteq R_i$. We say that V is a *legal shortcut set* if every $(R_i, Q_i), (R_j, Q_j) \in V$, satisfy $(R_i \setminus Q_i) \cap (R_j \setminus Q_j) = \emptyset$. By the preceding discussion, all pairs in a legal shortcut set can be shortcut simultaneously.

We say that a set S of nodes in the tree T has the *least common ancestor property* if S contains the root of T, and for every two nodes in S, their least common ancestor is also in S. A set S with the least common ancestor property induces a legal shortcut set in the following way: Let $R, Q \in S$ such that R is ancestor of Q, and the branch leading from R to Q has no other element of S. Define R_Q as the first son of R in the branch leading to Q. The shortcut set contains the pairs (R_Q, Q) for all such R and Q. It is easy to see that if (R_Q, Q) and $(R'_{Q'}, Q')$ are included in the set, then indeed $(R_Q \setminus Q) \cap (R'_{Q'} \setminus Q') = \emptyset$. Any non-empty intersection can result only from a situation in which there exists a node v in the tree that is below R and is an ancestor of both Q and R'. Such a situation contradicts the common ancestor property of S. We define $ShortCut(S)$ as the simultaneous application of $ShortCut(R_Q, Q)$ for all pairs in the induced legal shortcut set.

Lemma 5: *Let T be a tree, and S a set of nodes in T with the least common ancestor property. Suppose that for every $v \notin S$ there is an ancestor $w \in S$ such that the length of the path from w to v is at most d. Let T' be the modified tree, resulting from applying $ShortCut(S)$ to the tree T. Then the length of every branch in T' is at most $2 \cdot |S| + d$.*

Proof: Take an arbitrary branch in T'. Note that after applying $ShortCut(R_Q, Q)$ to T, the nodes R and Q are in distance 2 in T'. Consider the first pair of adjacent nodes on this branch which do not correspond to nodes in S. By the construction of the shortcut set, it follows that below this pair, there are no shortcut pairs. Therefore the length below this pair is at most d. Thus the longest branch in T' can be of length at most $2 \cdot |S| + d$. $\square$

We now give a method to construct a set S with the least common ancestor property, in a way which balances its size, $|S|$, with the distance d.

Lemma 6: *Let d be an integer ($d \geq 2$). Given a decomposition tree T with M nodes, there is a least common ancestor set S satisfying $|S| \leq \frac{2M}{d}$, such that every $v \notin S$ has an ancestor $w \in S$ which is at most d nodes above v.*

Proof: We construct S in two stages. In the first stage we iterate adding one node to S and remove a subset of nodes from consideration. For each node R of T, let $w(R)$ denote the number of descendants of R. While the tree has at least d nodes, it is possible to find a node with $w(R) \geq d$, but $W(Q) < d$ for every son Q of R. We add R to S, remove the subtree rooted at R, and repeat the process until less than d nodes are left.
In the second stage we close the set S under the least common ancestor of pairs, and finally add the root of T to S.
Denote by $S_T \subset T$ the resulting set. Clearly S_T is a least common ancestor set. The number of nodes added to S_T in the first stage cannot exceed $\frac{M}{d}$, since each such node resulted in the removal of at least d nodes from the set. Together with the nodes added in the second stage, S_T can be viewed as a tree. The second stage nodes correspond to internal nodes of degree ≥ 2 in this tree. Thus the number of the second stage nodes does not exceed the number of the first stage nodes. The total is bounded by

$$|S_T| \leq \frac{2M}{d}.$$

$\square$

The final step in the proof of Theorem 5 is to show how to choose (sequentially) k least common ancestor sets $S_1, S_2, \ldots, S_k$. Each of these sets yields one shortcut at the cost of revealing one additional bit, so overall k bits of additional information are revealed. Each set transforms the original protocol and its decomposition tree. Denote $M_1 = M, T_1 = T$ and $S_1 = S_{T_1}$, where the parameter d_1 is taken as $d_1 = M^{1-\frac{1}{k+1}}$. Removing the nodes of S_1 from T_1, we get a forest of disjoint trees, each having fewer than d_1 nodes. We apply the construction again to each of these subtrees $T_2^{(j)}$ independently, choosing $d_2 = M^{1-\frac{2}{k+1}}$.
In the i-th step we apply the construction with parameter $d_i = M^{1-\frac{i}{k+1}}$ to a collection of disjoint trees, each of size $M_i < d_{i-1} = M^{1-\frac{i-1}{k+1}}$. The length of any branch in the final tree is at most $\sum_{i=1}^{k} 2\,|S_{T_i}| \; + d_k$. This is the case since any branch which leaves the set S_i does not go back to

it. Thus, the worst case scenario is a branch going through all S_1 nodes, then through all S_2 nodes etc., and finally (after going through S_k nodes) going through at most d_k "free" nodes.
By the choice of the d_i's we have

$$|S_i| \leq \frac{2M_i}{d_i} < \frac{2d_{i-1}}{d_i} = 2M^{\frac{1}{k+1}}.$$

Therefore the length of any branch in the final tree is bounded above by

$$\sum_{i=1}^{k} 2\,|S_{T_i}| \quad + d_k = \sum_{i=1}^{k} 4M^{\frac{1}{k+1}} \quad + M^{\frac{1}{k+1}} = (4k+1)\cdot M^{\frac{1}{k+1}} \leq (4k+1)\cdot 2^{\frac{2n+1}{k+1}}.$$

This completes the proof of the theorem. □

7 Information-Theoretic Measures of Additional Information – $\mathcal{I}_{\rm i}$ and $\mathcal{I}_{\rm c\text{-}i}$

So far, we considered only one measure of privacy-cost: the combinatorial measure $\mathcal{I}_{\rm c}$. Next, we investigate two other measures: an information-theoretic measure $\mathcal{I}_{\rm i}$, and a measure $\mathcal{I}_{\rm c\text{-}i}$ that has both combinatorial and information-theoretic aspects. Each of the measures has a deterministic counterpart ($\mathcal{I}_{\rm i}^{\rm det}$ and $\mathcal{I}_{\rm c\text{-}i}^{\rm det}$), denoting the amount of information that must be divulged according to the measure if the communicators are restricted to deterministic protocols. $\mathcal{I}_{\rm c}^{\rm det}$ was defined in Section 3, and in Section 4 it is shown that $\mathcal{I}_{\rm c}^{\rm det}$ and $\mathcal{I}_{\rm c}$ are essentially the same.

As before, let $\mathcal{F}_n$ be the set of Boolean functions defined over $\{0,1\}^n \times \{0,1\}^n$. Corollary 2 shows that for most functions in $\mathcal{F}_n$:

- $\mathcal{I}_{\rm c}^{\rm det}$ (and therefore $\mathcal{I}_{\rm c}$) is about n;

We now show that for most functions in $\mathcal{F}_n$:

- $\mathcal{I}_{\rm c\text{-}i}^{\rm det}$ is about n while $\mathcal{I}_{\rm c\text{-}i}$ is about $n/2$.
- both $\mathcal{I}_{\rm i}^{\rm det}$ and $\mathcal{I}_{\rm i}$ are about $n/2$;

We partition this section in two. In the first subsection, we consider the deterministic measures. In subsection 7.2, we discuss randomized protocols. At the cost of some repetition, this enables us to provide a clearer, more intuitive, exposition, and to point out the differences between the different measures. We assume that the reader is familiar with basic information-theoretic quantities, such as entropy and mutual information. For details, see [5, 11].

7.1 Deterministic Protocols

Let ϕ and h be functions defined over $\{0,1\}^n \times \{0,1\}^n$. We say that ϕ is *determined* by h and x if for all $x \in \{0,1\}^n$ and $y_1, y_2 \in \{0,1\}^n$,

$$h(x, y_1) = h(x, y_2) \quad \text{imply} \quad \phi(x, y_1) = \phi(x, y_2). \tag{3}$$

We similarly say that ϕ is determined by h and y if a party that knows y and $h(x, y)$ can infer $\phi(x, y)$. A deterministic protocol on $\{0,1\}^n \times \{0,1\}^n$ assigns a sequence of transmitted bits to every input $(x, y) \in \{0,1\}^n \times \{0,1\}^n$ and can therefore be viewed as a function over $\{0,1\}^n \times \{0,1\}^n$. In that sense, we redefine $\mathcal{I}_{\mathrm{c}}^{\mathrm{det}}$:

$$\mathcal{I}_{\mathrm{c}}^{\mathrm{det}}(f) \stackrel{\mathrm{def}}{=} \min\{\log|\mathrm{range}(h)| : \quad \text{there exists a deterministic protocol for } f \text{ that is determined both by } h \text{ and } x \text{ and by } h \text{ and } y\}\ . \tag{4}$$

While $\mathcal{I}_{\mathrm{c}}^{\mathrm{det}}$ is well defined even if no probability distribution on the input space is assumed, the definitions of $\mathcal{I}_{\mathrm{i}}^{\mathrm{det}}$ and $\mathcal{I}_{\text{c-i}}^{\mathrm{det}}$ implicitly assume an underlying distribution. We view x and y as a pair (X, Y) of random variables drawn from some a-priori distribution which is known to both parties. $P_{\mathcal{Y}}$ knows Y, hence the amount of information he obtains about X during the execution of a protocol $\mathcal{A}$ is the conditional mutual information $I(X; \mathcal{A}(X,Y)|Y)$. Similarly, $\mathcal{A}$ provides $P_{\mathcal{X}}$ with $I(Y; \mathcal{A}(X,Y)|X)$ bits of information about Y. Every Boolean function has a deterministic protocol computing it in which $P_{\mathcal{X}}$ learns at most one bit of information: $P_{\mathcal{X}}$ transmits X, then $P_{\mathcal{Y}}$ responds with the value of the function. Likewise, every Boolean function has a protocol that provides $P_{\mathcal{Y}}$ with at most one bit of information. We will show that for some functions, all protocols provide at least one of $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ with a lot of information. Define

$$I_{\mathcal{A}}(X,Y) \stackrel{\mathrm{def}}{=} \max\{I(X; \mathcal{A}(X,Y)|Y)\ ,\ I(Y; \mathcal{A}(X,Y)|X)\}$$

to be the maximum of the information that $\mathcal{A}$ gives $P_{\mathcal{Y}}$ and $P_{\mathcal{X}}$.

To define $\mathcal{I}_{\mathrm{i}}^{\mathrm{det}}$, let

$$I_f^{\mathrm{det}}(X,Y) \stackrel{\mathrm{def}}{=} \min\{I_{\mathcal{A}}(X,Y) : \mathcal{A} \text{ is a deterministic protocol computing } f\} \tag{5}$$

be the minimum, over all deterministic protocols computing f, of the information they divulge. (Note that, as X and Y range over a finite range, there is a finite number of protocols so the minimum is well defined.) We then define

$$\mathcal{I}_{\mathrm{i}}^{\mathrm{det}}(f) \stackrel{\mathrm{def}}{=} \sup\{I_f^{\mathrm{det}}(X,Y) : (X,Y) \text{ is a random pair distributed over } \{0,1\}^n \times \{0,1\}^n\ \}$$

to be the amount of information that at least one communicator must learn about the other's random variable if they compute f using a deterministic protocol, maximized over all possible underlying distributions of (X, Y).

To define $\mathcal{I}_{\text{c-i}}^{\text{det}}$, let

$$I_{\mathcal{A}} \stackrel{\text{def}}{=} \sup\{I_{\mathcal{A}}(X,Y) : (X,Y) \text{ is a random pair distributed over } \{0,1\}^n \times \{0,1\}^n \},$$

namely, the maximum of $I_{\mathcal{A}}(X,Y)$ over all possible probability distributions underlying (X,Y). We then define

$$\mathcal{I}_{\text{c-i}}^{\text{det}}(f) \stackrel{\text{def}}{=} \min\{I_{\mathcal{A}} : \mathcal{A} \text{ is a deterministic protocol computing } f\}. \tag{6}$$

Both $\mathcal{I}_{\text{i}}^{\text{det}}$ and $\mathcal{I}_{\text{c-i}}^{\text{det}}$ are defined in information-theoretic terms, however, $\mathcal{I}_{\text{c-i}}^{\text{det}}$ is more combinatorial in nature. For $\mathcal{I}_{\text{c-i}}^{\text{det}}(f)$ to be low, f must have a single protocol that divulges little information regardless of the probability distribution underlying (X,Y). For $\mathcal{I}_{\text{i}}^{\text{det}}(f)$ to be low, it suffices that for every distribution there is a protocol divulging little information. $\mathcal{I}_{\text{i}}^{\text{det}}$ is more pertinent when the probability distribution underlying (X,Y) is known before the communication protocol is designed. $\mathcal{I}_{\text{c-i}}^{\text{det}}$ is more applicable when the underlying distribution is not known, and in that respect, is more closely related to the combinatorial measure $\mathcal{I}_{\text{c}}^{\text{det}}$. The next Lemma provides a geometric interpretation of $\mathcal{I}_{\text{c-i}}^{\text{det}}$, thereby further justifying the notation.

Recall that a *rectangle* is a Cartesian-product in $\{0,1\}^n \times \{0,1\}^n$, and that every deterministic protocol $\mathcal{A}$ induces a partition $\mathcal{X}_1 \times \mathcal{Y}_1, \ldots, \mathcal{X}_k \times \mathcal{Y}_k$ of $\{0,1\}^n \times \{0,1\}^n$ into rectangles. For each $x \in \{0,1\}^n$ let

$$\ell_{\text{row}}(x) \stackrel{\text{def}}{=} |\{i \in \{1,\ldots,k\} : x \in \mathcal{X}_i\}| \tag{7}$$

be the number of rectangles induced by $\mathcal{A}$ that intersect row x. Similarly, for $y \in \{0,1\}^n$, let $\ell_{\text{col}}(y)$ be the number of rectangles induced by $\mathcal{A}$ that intersect column y. Let

$$\ell_{\text{row}} \stackrel{\text{def}}{=} \max\{\ell_{\text{row}}(x) : x \in \{0,1\}^n\}$$

be the maximum number of rectangles intersecting a row and define ℓ_{col} similarly. Finally, define

$$\ell_{\mathcal{A}} \stackrel{\text{def}}{=} \max\{\ell_{\text{row}}, \ell_{\text{col}}\}$$

to be the maximum number of rectangles in any row or column.

Lemma 7: *(Geometric interpretation of $I_{\mathcal{A}}$ and $\mathcal{I}_{c\text{-}i}^{det}$.)*
A. For all deterministic protocols $\mathcal{A}$,

$$I_{\mathcal{A}} = \log \ell_{\mathcal{A}} \ .$$

B. For all $f \in \mathcal{F}_n$,

$$\mathcal{I}_{c\text{-}i}^{det}(f) = \min\{\log \ell_{\mathcal{A}} : \mathcal{A} \textit{ is a deterministic protocol computing } f\}.$$

Proof: Part (B) follows from (A) and the definition of $\mathcal{I}^{\mathrm{det}}_{\mathrm{c\text{-}i}}(f)$; we prove (A). For every deterministic protocol $\mathcal{A}$, every distribution (X, Y) and every $x \in \{0,1\}^n$,

$$I(Y; \mathcal{A}(X,Y)|x) \leq H(\mathcal{A}(X,Y)|x) \leq \log \ell_{\mathrm{row}}(x) \leq \log \ell_{\mathcal{A}} \ ,$$

hence $I(Y; \mathcal{A}(X,Y)|X) \leq \log \ell_{\mathcal{A}}$. Similarly $I(X; \mathcal{A}(X,Y)|Y) \leq \log \ell_{\mathcal{A}}$, and $I_{\mathcal{A}} \leq \log \ell_{\mathcal{A}}$ follows. To obtain the reverse inequality, find a row or a column in $\{0,1\}^n$ intersecting $\ell_{\mathcal{A}}$ rectangles. Pick $\ell_{\mathcal{A}}$ points in this row (or column), one from each rectangle, and let (X, Y) be distributed uniformly over these points. $\square$

In Corollary 4, we will use this interpretation to prove that most functions $f \in \mathcal{F}_n$ have $\mathcal{I}^{\mathrm{det}}_{\mathrm{c\text{-}i}}(f) \geq n - \log n - 2$. But first, we relate the three measures.

Theorem 6*: For all functions f,*

$$\mathcal{I}^{det}_{i}(f) \leq \mathcal{I}^{det}_{c\text{-}i}(f) \leq \mathcal{I}^{det}_{c}(f) \ .$$

Proof: The first inequality follows from the definition of $\mathcal{I}^{\mathrm{det}}_{\mathrm{i}}$ as a "max min" and of $\mathcal{I}^{\mathrm{det}}_{\mathrm{c\text{-}i}}$ as a "min max." To prove the second inequality, let h be a function achieving $\mathcal{I}^{\mathrm{det}}_{\mathrm{c}}(f)$ in Definition (4) and let $\mathcal{A}$ be a deterministic protocol for f which is determined both by h and y and by h and x. As $\mathcal{A}$ is determined by h and x, all distributions (X, Y) satisfy:

$$I(Y; \mathcal{A}(X,Y)|X) \leq I(Y; h(X,Y)|X) \ . \tag{8}$$

But,

$$\begin{aligned} I(Y; h(X,Y)|X) &\leq H(h(X,Y)|X) \\ &\leq H(h(X,Y)) \\ &\leq \log|\mathrm{Range}(h)| \\ &= \mathcal{I}^{\mathrm{det}}_{\mathrm{c}}(f), \end{aligned}$$

and, similarly, $I(X; \mathcal{A}(X,Y)|Y) \leq \mathcal{I}^{\mathrm{det}}_{\mathrm{c}}(f)$. Hence,

$$I_{\mathcal{A}}(X,Y) \leq \mathcal{I}^{\mathrm{det}}_{\mathrm{c}}(f) \ .$$

This holds for all distributions (X, Y), hence

$$I_{\mathcal{A}} \leq \mathcal{I}^{\mathrm{det}}_{\mathrm{c}}(f) \ .$$

The theorem follows as $\mathcal{I}^{\mathrm{det}}_{\mathrm{c\text{-}i}}(f)$ is the minimum of $I_{\mathcal{A}}$ over all deterministic protocols $\mathcal{A}$ computing f. $\square$

We proceed to show that for most functions $f \in \mathcal{F}_n$, $\mathcal{I}^{\mathrm{det}}_{\mathrm{c\text{-}i}}(f)$ is roughly n while $\mathcal{I}^{\mathrm{det}}_{\mathrm{i}}(f)$ is about $\frac{n}{2}$. As with $\mathcal{I}_{\mathrm{C}}$ (Theorem 3), we do so by relating $\mathcal{I}^{\mathrm{det}}_{\mathrm{c\text{-}i}}(f)$ and $\mathcal{I}^{\mathrm{det}}_{\mathrm{i}}(f)$ to r_f — the largest width of an f-constant rectangle.

Theorem 7: *For all functions $f \in \mathcal{F}_n$,*

$$\mathcal{I}_{c\text{-}i}^{det}(f) \geq n - \log r_f - 1 \ .$$

Proof: We show that in every decomposition of $\{0,1\}^n \times \{0,1\}^n$ into f-constant rectangles, there is either a row or a column that intersects at least $\frac{2^n}{2r_f}$ rectangles. Lemma 7 then implies the result.

Let $\mathcal{R} = \{R_1, \ldots, R_k\}$ be a decomposition of $\{0,1\}^n \times \{0,1\}^n$ into f-constant rectangles where $R_i = \mathcal{X}_i \times \mathcal{Y}_i$. By definition, $\min\{|\mathcal{X}_i|, |\mathcal{Y}_i|\} \leq r_f$ for all $i \in \{1, \ldots, k\}$. Recall that $\ell_{\text{row}}(x)$ is the number of rectangles intersecting row x, and that $\ell_{\text{col}}(y)$ is the number of rectangles intersecting column y. Then,

$$\begin{aligned}
\sum_{x \in \{0,1\}^n} \ell_{\text{row}}(x) + \sum_{y \in \{0,1\}^n} \ell_{\text{col}}(y) &= \sum_{i=1}^{k} (|\mathcal{X}_i| + |\mathcal{Y}_i|) \\
&\geq \frac{1}{r_f} \sum_{i=1}^{k} r_f \cdot \max\{|\mathcal{X}_i|, |\mathcal{Y}_i|\} \\
&\geq \frac{1}{r_f} \sum_{i=1}^{k} \min\{|\mathcal{X}_i|, |\mathcal{Y}_i|\} \cdot \max\{|\mathcal{X}_i|, |\mathcal{Y}_i|\} \\
&= \frac{1}{r_f} \sum_{i=1}^{k} |\mathcal{X}_i| \cdot |\mathcal{Y}_i| \\
&= \frac{2^{2n}}{r_f} \ .
\end{aligned}$$

The total number of x's and y's is 2^{n+1}, hence there exists at least one row or one column which intersects at least $\frac{2^{2n}}{2 \cdot r_f \cdot 2^n} = \frac{2^n}{2r_f}$ rectangles. $\square$

We can now prove that $\mathcal{I}_{\text{c-i}}^{\text{det}}$ is high for "most" functions and for functions with a Hadamard-type function matrices. As in Section 5, let

$$\mathcal{F}_n^* \stackrel{\text{def}}{=} \{f \in \mathcal{F}_n : r_f < 2n\}$$

be the set of functions in $\mathcal{F}_n$ whose monochromatic squares all have width $< 2n$. Lemma 3 showed that

$$|\mathcal{F}_n^*| \geq \left(1 - \frac{2}{[(2n)!]^2}\right) |\mathcal{F}_n| \ .$$

Hence,

Corollary 4: *For most functions $f \in \mathcal{F}_n$,*

$$\mathcal{I}_{c\text{-}i}^{det}(f) \geq n - \log n - 2 \ .$$

$\square$

Invoking Lindsey's Lemma (See Corollary 3), we obtain:

Corollary 5: *Let $f \in \mathcal{F}_n$. If M_f is of Hadamard type, then*

$$\mathcal{I}_c^{det} \geq \mathcal{I}_{c\text{-}i}^{det}(f) \geq n - \log n - 2 \ .$$

$\square$

Similar results hold for $\mathcal{I}_{\text{i}}$.

Theorem 8: *For all functions $f \in \mathcal{F}_n$,*

$$\mathcal{I}_i^{det}(f) \geq \frac{n - \log r_f}{2} \ .$$

Proof: Let (X, Y) be distributed uniformly over $\{0,1\}^n \times \{0,1\}^n$. We prove the lemma by showing that every deterministic protocol $\mathcal{A}$ for f satisfies

$$I(Y; \mathcal{A}(X,Y)|X) + I(X; \mathcal{A}(X,Y)|Y) \geq n - \log r_f \ .$$

Let $\mathcal{A}$ be a deterministic protocol for f. Recall that for every sequence s of bits,

$$\mathcal{A}^{-1}(s) \stackrel{\text{def}}{=} \{(x,y) : \mathcal{A}(x,y) = s\}$$

is the set of inputs for which $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ exchange the sequence s. As mentioned in Section 2, $\mathcal{A}^{-1}(s)$ is an f-constant rectangle for all s. For a rectangle $R = A \times B$, Let $b(R) = |B|$ be its 'base,' and let $h(R) = |A|$ be its 'height.' Abbreviating the event $\mathcal{A}(X,Y) = s$ as s, we have:

$$\begin{aligned}
H(Y|\mathcal{A}(X,Y), X) + H(X|\mathcal{A}(X,Y), Y) &= \sum_s Pr(s)\,(H(Y|s,X) + H(X|s,Y)) \\
&\leq \sum_s Pr(s)\left(\log b(\mathcal{A}^{-1}(s)) + \log h(\mathcal{A}^{-1}(s))\right) \\
&\leq \sum_s Pr(s)(\log r_f + \log 2^n) \\
&= n + \log r_f \ .
\end{aligned}$$

Therefore,

$$\begin{aligned}
I(Y; \mathcal{A}(X,Y)|X) &+ I(X; \mathcal{A}(X,Y)|Y) \\
&= H(Y|X) + H(X|Y) - H(Y|\mathcal{A}(X,Y), X) - H(X|\mathcal{A}(X,Y), Y) \\
&\geq 2n - n - \log r_f \\
&= n - \log r_f \ .
\end{aligned}$$

$\square$

We note that the derivation above could have been somewhat shortened using the identity $I(Y; \mathcal{A}(X,Y)|X) = H(\mathcal{A}(X,Y)|X)$ and its symmetric counterpart. However, this equation does not generalize to randomized protocols, whereas the ones used here do.

Corollary 6*:*

A. *For all functions* $f \in \mathcal{F}_n^*$*,*

$$\mathcal{I}_i^{det}(f) \geq \frac{n - \log n - 1}{2} .$$

B. *For all functions* $f \in \mathcal{F}_n$*,*

$$\mathcal{I}_i^{det}(f) \leq \frac{n}{2} + 2 .$$

Proof: Part (A) follows from Theorem 8 and the definition of $\mathcal{F}_n^*$. To prove part (B), let

$$\alpha \stackrel{\text{def}}{=} \max\{\max_{x\in\{0,1\}^n}\{\sum_{y\in\{0,1\}^n} Pr(x,y)\}, \max_{y\in\{0,1\}^n}\{\sum_{x\in\{0,1\}^n} Pr(x,y)\}\}$$

be the maximum probability of any row or column. Without loss of generality assume that α is the probability of row x_0. Fix a partition A, B of $\{0,1\}^n$ such that $x_0 \in B$ and

$$\frac{1}{2} - \frac{\alpha}{2} \leq Pr(A) \leq Pr(B) \leq \frac{1}{2} + \frac{\alpha}{2}$$

where $Pr(A)$ denotes the probability that X is in A, similarly for $Pr(B)$. The protocol $\mathcal{A}$ proceeds as follows. If $X \in B$, then $P_\mathcal{X}$ transmits 0, followed by the sequence X. If $X \in A$, then $P_\mathcal{X}$ transmits 1, and then $P_\mathcal{Y}$ transmits the sequence Y.

The amount of information divulged to $P_\mathcal{X}$ is given by:

$$\begin{aligned}
I(Y;\Phi(X,Y)|X) &= \sum_{x\in\{0,1\}^n} Pr(x) I(Y;\mathcal{A}(X,Y)|x) \\
&= \sum_{x\in A} Pr(x) I(Y;\mathcal{A}(X,Y)|x) + \sum_{x\in B} Pr(x) I(Y;\mathcal{A}(X,Y)|x) \\
&\leq \sum_{x\in A} Pr(x)\cdot n + \sum_{x\in B} Pr(x)\cdot 0 \\
&\leq Pr(A)\cdot n \\
&\leq n/2 .
\end{aligned}$$

For the amount of information obtained by $P_\mathcal{Y}$, let Z be the random variable whose value is A if $X \in A$ and B if $X \in B$. The variable Z is determined by X. Abbreviating the event $Z = A$ as A and $Z = B$ as B, we have:

$$\begin{aligned}
& I(X;\mathcal{A}(X,Y)|Y) \\
= {} & H(X|Y) - H(X|\mathcal{A}(X,Y),Y) \\
= {} & H(Z|Y) + H(X|Z,Y) - H(Z|\mathcal{A}(X,Y),Y) - H(X|Z,\mathcal{A}(X,Y),Y) \\
= {} & H(Z) + Pr(A)H(X|A,Y) + Pr(B)H(X|B,Y) - 0 - Pr(A)H(X|A,Y) - Pr(B)\cdot 0 \\
= {} & H(Z) + Pr(B)H(X|B,Y) .
\end{aligned}$$

But,

$$
\begin{aligned}
H(X|B,Y) &= \sum_{y\in\{0,1\}^n} Pr(y)H(X|B,y) \\
&= \sum_{y\in\{0,1\}^n} Pr(y)\left(\sum_{x\in\{0,1\}^n} Pr(x|B,y)\log\frac{1}{Pr(x|B,y)}\right) \\
&= \sum_{y\in\{0,1\}^n} Pr(y)\left(Pr(x_0|B,y)\log\frac{1}{Pr(x_0|B,y)} + \sum_{x\neq x_0} Pr(x|B,y)\log\frac{1}{Pr(x|B,y)}\right) \\
&\leq \sum_{y\in\{0,1\}^n} Pr(y)\left(1 + Pr(X\neq x_0|B,y)\sum_{x\neq x_0}\frac{Pr(x|B,y)}{Pr(X\neq x_0|B,y)}\log\frac{1}{Pr(x|B,y)}\right) \\
&\leq \sum_{y\in\{0,1\}^n} Pr(y)\left(1 + Pr(X\neq x_0|B,y)\cdot n\right) \\
&= 1 + Pr(X\neq x_0|B) \\
&\leq 1 + (1-\alpha)\cdot n\ .
\end{aligned}
$$

Incorporating this into the previous inequality, we obtain

$$
\begin{aligned}
I(X;\mathcal{A}(X,Y)|Y) &= H(Z) + Pr(B)H(X|B,Y) \\
&\leq 1 + \left(\frac{1}{2}+\frac{\alpha}{2}\right)\cdot(1+(1-\alpha)\cdot n) \\
&\leq 2 + (1-\alpha^2)\cdot\frac{n}{2} \\
&\leq 2 + \frac{n}{2}\ .
\end{aligned}
$$

and part (B) follows. □

7.2 Randomized Protocols

To extend $\mathcal{I}_{\text{c}}^{\text{det}}$, $\mathcal{I}_{\text{c-i}}^{\text{det}}$, and $\mathcal{I}_{\text{i}}^{\text{det}}$ to encompass randomized protocols, we need to generalize two definitions made in the last subsection. A randomized protocol Φ induces a probability distribution $\Phi(x,y)$ on the bits communicated when $P_{\mathcal{X}}$ has x and $P_{\mathcal{Y}}$ has y. We therefore view $\Phi(x,y)$ as a random variable. Generalizing definition (3), let Φ and h be functions defined over $\{0,1\}^n\times\{0,1\}^n$ where, this time, the function Φ may be random, i.e., $\Phi(x,y)$ may be a random variable for some, or all, (x,y). We say that Φ is *determined by* h *and* x if for all $x\in\{0,1\}^n$ and all $y_1,y_2\in\{0,1\}^n$,

$$h(x,y_1)=h(x,y_2) \quad \text{implies} \quad \Phi(x,y_1)=\Phi(x,y_2),$$

namely, the probability distribution of the random variable $\Phi(x,y)$ is determined by x and $h(x,y)$. Equivalently, for all values x,y,h, and s of $X,Y,h(X,Y)$, and $\Phi(X,Y)$,

$$Pr(s|x,y,h) = Pr(s|x,h)\ .$$

We then define the measures $\mathcal{I}_{\text{c}}$, $\mathcal{I}_{\text{i}}$, and $\mathcal{I}_{\text{c-i}}$ just as we defined their deterministic counterparts $\mathcal{I}_{\text{c}}^{\text{det}}$, $\mathcal{I}_{\text{i}}^{\text{det}}$, and $\mathcal{I}_{\text{c-i}}^{\text{det}}$, except that the minimization in Equations (4), (5), and (6) is performed over all protocols, not just deterministic ones.

The following observation will be useful in generalizing the proofs given for deterministic protocols to arbitrary ones.

Lemma 8*: Φ is determined by h and x if and only if Y and $\Phi(X,Y)$ are mutually independent given X and $h(X,Y)$.*

Proof: Let x, y, h, and s be the values of $X, Y, h(X,Y)$, and Φ. Then,

$$\frac{Pr(y,s|x,h)}{Pr(y|x,h)\cdot Pr(s|x,h)} = \frac{Pr(s|y,x,h)}{Pr(s|x,h)} \; .$$

□

We first prove the analog of Theorem 6.

Theorem 9*: For all functions f,*

$$\mathcal{I}_i(f) \le \mathcal{I}_{c\text{-}i}(f) \le \mathcal{I}_c(f) \; .$$

Proof: As in Theorem 6, the first (left) inequality follows from the definition. To prove the second inequality, let h be a function achieving $\mathcal{I}_{\text{c}}(f)$ in Definition 4 and let Φ be a protocol for f that is determined by h and y and by h and x. Lemma 8 observed that X and $\Phi(X,Y)$ are mutually independent given Y and $h(X,Y)$. Hence, by the Data-Processing Inequality, cf., Theorem 1.5 in [11], we again obtain Inequality (8):

$$I(Y;\Phi(X,Y)|X) \le I(Y;h(X,Y)|X)$$

and the proof continues as in Theorem 6. □

Clearly, allowing randomized protocols cannot increase the amount of information that must be divulged during the computation of f. However, for $\mathcal{I}_{\text{c}}^{\text{det}}$, randomized protocols cannot reduce that quantity either, as it follows from Lemma 2 that for all functions f, $\mathcal{I}_{\text{c}}(f) = \mathcal{I}_{\text{c}}^{\text{det}}(f)$.

Corollary 4 showed that for most functions in $\mathcal{F}_n$, $\mathcal{I}_{\text{c}}^{\text{det}}$ is about n, and this does not change when we consider $\mathcal{I}_{\text{c}}$ instead. Theorem 8 showed that for most functions in $\mathcal{F}_n$, $\mathcal{I}_{\text{i}}^{\text{det}}$ is about $n/2$ and the next lemma implies that this does not change when we consider $\mathcal{I}_{\text{i}}$ instead. However, the next lemma shows that for most functions in $\mathcal{F}_n$, $\mathcal{I}_{\text{c-i}}$ is also about $n/2$, whereas Corollary 4 showed that $\mathcal{I}_{\text{c-i}}^{\text{det}}$ is about n.

Theorem 10*: A. For all functions $f \in \mathcal{F}_n$,*

$$\mathcal{I}_i(f) \le \mathcal{I}_{c\text{-}i}(f) \le \frac{n+1}{2} \; .$$

B. For most functions $f \in \mathcal{F}_n$,

$$\frac{n - \log n - 1}{2} \leq \mathcal{I}_i(f) \leq \mathcal{I}_{c\text{-}i}(f) \ .$$

Proof: A. The first inequality is by Theorem 9. For the second inequality, we present the following protocol Φ which has $I_\Phi \leq \frac{n+1}{2}$: $P_\mathcal{X}$ flips a fair coin and transmits its 0-1 outcome C to $P_\mathcal{Y}$. If C is 0, $P_\mathcal{Y}$ transmits y and $P_\mathcal{X}$ computes and transmits $f(x,y)$. If C is 1, they reverse the roles. I.e., $P_\mathcal{X}$ transmits x and $P_\mathcal{Y}$ computes and transmits $f(x,y)$. As C is independent of X, we have:

$$\begin{aligned} I(X;\Phi(X,Y)|Y) &\leq I(X;\Phi(X,Y)|Y,C) \\ &= \sum_{c\in\{0,1\}} Pr(c) I(X;\Phi(X,Y)|Y,c) \\ &\leq \frac{1}{2}\cdot 1 + \frac{1}{2}\cdot n \\ &= \frac{n+1}{2} \ . \end{aligned}$$

and the symmetric argument shows that $I(Y;\Phi(X,Y)|X) \leq \frac{n+1}{2}$.

B. For every input $(x,y) \in \{0,1\}^n \times \{0,1\}^n$, the protocol Φ defines a probability distribution on the sequences $\Phi(x,y)$ of transmitted bits. Let s be a sequence of bits. Define

$$\Phi(s) \stackrel{\text{def}}{=} \{(x,y) : \Phi(x,y) \text{ can assume the value } s\}$$

to be the set of inputs for which Φ may exchange s. As for deterministic protocols, it is easy to see that if Φ is error-free for f, then for all s, $\Phi(s)$ is an f-monochromatic rectangle. The quantity $Pr(s)$ now depends not only on the input distribution (X,Y), but also on the randomization of the protocol. However, the proof proceeds as in Theorem 8. □

It is possible to further extend the class of randomized protocols by considering protocols which might err. We say that a protocol $\mathcal{A}$ *computes f with ε advantage* if

$$\forall x,y \in \{0,1\}^n : Pr(L_\mathcal{A}(x,y) = f(x,y)) \geq \frac{1}{2} + \varepsilon \ .$$

The quantities $\mathcal{I}^\varepsilon_{\text{c}}$, $\mathcal{I}^\varepsilon_{\text{c-i}}$, and $\mathcal{I}^\varepsilon_{\text{i}}$ are defined analogously, where the protocols considered are ε advantage protocols. While Lemma 2 implies that $\mathcal{I}^\varepsilon_{\text{c}} = \mathcal{I}^{\text{det}}_{\text{c}}$, the other two measures satisfy, for every Boolean function f,

$$\mathcal{I}^\varepsilon_{\text{i}}(f) \leq \mathcal{I}^\varepsilon_{\text{c-i}}(f) \leq \varepsilon \cdot n + 1 \ .$$

This upper bound follows from the following protocol: $P_\mathcal{X}$ flips a coin which is $1-2\varepsilon$ biased towards heads. He notifies $P_\mathcal{Y}$ of the outcome. If this coin came up heads, then $P_\mathcal{X}$ flips an unbiased coin, and its result is the final outcome of the protocol. If the original coin comes up tails, then the two parties proceed as in the protocol of Theorem 10, part (A). Therefore by allowing errors, both $\mathcal{I}^\varepsilon_{\text{i}}(f)$ and $\mathcal{I}^\varepsilon_{\text{c-i}}(f)$ can be substantially reduced.

8 Concluding Remarks

The notion of privately computable functions can be restated in terms of simulations. A function is privately computable if it has a protocol where each party can *exactly* simulate the communication given its input and the value of the function. In this form, privacy strongly resembles the notion of *perfect zero knowledge* as defined by Goldwasser, Micali, and Rackoff [9].

Protocols computing non-private functions can also be described in the language of simulations. A protocol for is h-private if each party can simulate the communication exactly, given its input and the value of h.

In the early version of [9] there is an attempt to define knowledge complexity quantitatively, in terms of probabilities of successful simulations. This formulation seems to be ad-hoc and unnatural [7]. An alternative formulation can be based on the approach presented here: A protocol reveals at most k bits of knowledge if its execution can be perfectly simulated given the input x and the value of a function $h(x)$, whose range has size $\leq 2^k$. Recently, Goldreich [7] independently proposed another formulation, which is more general than ours, and Goldreich and Petrank [8] suggested several definitions for the amount of knowledge revealed in an interactive proof system and studied the relations between these definitions. We believe that these definitions have the potential of yielding interesting results in the computationally bounded context.

Another interesting problem is to quantify more accurately the relations between $\mathcal{I}_{\mathrm{c}}$, $\mathcal{I}_{\mathrm{c\text{-}i}}$, and $\mathcal{I}_{\mathrm{i}}$. For example, is $\mathcal{I}_{\mathrm{c}}(f) \leq 2\mathcal{I}_{\mathrm{i}}(f)$ for every Boolean function f?

9 Acknowledgments

We thank Oded Goldreich for many stimulating discussions.

References

[1] Abadi M., J. Feigenbaum, and J. Kilian, "On Hiding Information from an Oracle", *JCSS*, Vol. 39, No. 1, 1989, pp. 21–50.

[2] Beaver, D., "Perfect Privacy for Two Party Protocols", Technical Report TR-11-89, Harvard University, 1989.

[3] Ben-or M., S. Goldwasser, and A. Wigderson, "Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation" *Proc. of 20th Annual ACM Symposium on Theory of Computing*, 1988, pp. 1-10.

[4] Chor, B., and E. Kushilevitz, "A Zero-One Law for Boolean Privacy", *SIAM Jour. on Disc. Math.*, Vol. 4, No. 1, February 1991, pp. 36-47.

[5] Cover, T., and J. Thomas, "Elements of Information Theory", John Wiley and Sons, 1991.

[6] Erdos P., and J. Spencer, *Probabilistic Methods in Combinatorics*, Academic Press, New York, 1974.

[7] Goldreich, O., "Foundations of Cryptography: List of References", Technical Report 573, Technion, 1989.

[8] Goldreich, O., and E. Petrank "Quantifying Knowledge Complexity", *Proc. of 32nd Annual Symposium on Foundations of Computer Science*, 1991, pp. 59–68.

[9] Goldwasser, S., S. Micali, and C. Rackoff, "The Knowledge Complexity of Interactive Proof Systems", *SIAM J. Comp.*, Vol. 18, No. 1, 1989, pp. 186–208. Earlier version in *Proc. of 17th Annual ACM Symposium on Theory of Computing*, 1985, pp. 291–304.

[10] Kushilevitz, E., "Privacy and Communication Complexity", *SIAM Jour. on Disc. Math.* Vol. 5, No. 2, May 1992, pp. 273-284.

[11] R.J. McEliece, "The Theory of Information and Coding", Encyclopedia of Mathematics and its Applications, Addison-Wesley Publishing Company, Reading MA, 1977.

[12] MacWilliams, F.J., and N.J.A. Sloane, *The Theory of Error–Correcting Codes*, North-Holland, Amsterdam, 1977.

[13] Yao, A. C., "Some Complexity Questions Related to Distributed Computing", *Proc. of 11th Annual ACM Symposium on Theory of Computing*, 1979.