A Zero – One Law for Boolean Privacy*

Benny Chor†
Eyal Kushilevitz

Department of Computer Science
Technion, Haifa 32000, Israel

Abstract

A Boolean function $f : A_1 \times A_2 \times \ldots \times A_n \rightarrow \{0,1\}$ is t – private if there exists a protocol for computing f so that no coalition of size $\leq t$ can infer any additional information from the execution, other than the value of the function. We show that f is $\lceil \frac{n}{2} \rceil$ – private if and only if it can be represented as

$$f(x_1, x_2, \ldots, x_n) = f_1(x_1) \oplus f_2(x_2) \oplus \ldots \oplus f_n(x_n) ,$$

where the f_i are arbitrary Boolean functions. It follows that if f is $\lceil \frac{n}{2} \rceil$ – private, then it is also n – private. Combining this with a result of Ben-Or, Goldwasser, and Wigderson, we derive an interesting "zero–one" law for private distributed computation of Boolean functions: Every Boolean function defined over a finite domain is either n – private, or it is $\lfloor \frac{n-1}{2} \rfloor$ – private but not $\lceil \frac{n}{2} \rceil$ – private.

We also investigate a weaker notion of privacy, where (a) coalitions are allowed to infer a limited amount of additional information, and (b) there is a probability of error in the final output of the protocol. We show that the same characterization of $\lceil \frac{n}{2} \rceil$ – private Boolean functions holds, even under these weaker requirements. In particular, this implies that for Boolean functions, the strong and the weak notions of privacy are equivalent.

*An early version of this paper appeared in *Proceedings of 21st Symposium on Theory of Computing*, ACM, 1989, pp. 62–72.

†Part of this research was done while visiting ICSI, Berkeley, CA.

1 Introduction

A set of n parties, each holding an input value x_i, wishes to distributively compute the value of a Boolean function $f(x_1, x_2, ..., x_n) \in \{0,1\}$. The participants communicate via a complete network of secure channels (no eavesdropping). The participants are honest – they send messages according to the prescribed protocol for f. However, honesty is no deterrent against curiosity. A subset of the participants (a coalition) might get together after the execution of the protocol and compare notes in an attempt to infer additional information on the inputs of non-coalition parties. Additional information is any information that does not follow from the value of the function and the inputs of the coalition parties. As an example of additional information consider a case where $f(0,0,\ldots,0,0,0) = f(0,0,\ldots,0,0,1)$ and, based on the execution of the protocol, the first $n/2$ participants can infer that the n-th input is more likely to be 1 than 0. In general, any information-theoretic advantage can be used by the coalition even if this requires, for example, exponential computational resources[1].

A function f is called t – *private* if there is a protocol for computing f so that no coalition of size $\leq t$ can get any additional information. The fundamental result in this area is due to Ben-Or, Goldwasser and Wigderson [BGW], who have shown that every n – variable function defined over a finite domain is $\left\lfloor \frac{n-1}{2} \right\rfloor$ – private. (A similar result was independently obtained by Chaum, Crepeau, and Damgard [CCD].) Ben-Or, Goldwasser and Wigderson also showed that *certain* functions (e.g. the OR function) are not $\left\lceil \frac{n}{2} \right\rceil$ – private, while *certain* others (e.g. the XOR function) are n – private. Other than these two examples, very little was known about t – private functions for $t \geq \left\lceil \frac{n}{2} \right\rceil$.

In this paper, we address the general problem of t – privacy in the range $\left\lceil \frac{n}{2} \right\rceil \leq t \leq n$. We raise two major questions:

1. What is the structure of the "privacy hierarchy"? Is it the case that for every t in the range $\left\lceil \frac{n}{2} \right\rceil \leq t < n$ there are functions that are t – private but not $t+1$ – private or does the hierarchy consists of two isolated levels $\left(\left\lfloor \frac{n-1}{2} \right\rfloor \text{ – privacy and } n \text{ – privacy} \right)$?

2. Is it possible to relate the form of a function f to its attainable privacy?

We resolve both questions for Boolean functions $f : A_1 \times A_2 \times ... \times A_n \rightarrow \{0,1\}$ defined over arbitrary (possibly infinite) domains. We give a complete characterization of Boolean functions that are $\left\lceil \frac{n}{2} \right\rceil$ – private. It is shown that every such function can be expressed as the exclusive-or of n Boolean functions, each depending on a single variable. There is a simple n – private randomized protocol [Bh] for computing functions of the form $f(x_1, x_2, ..., x_n) = f_1(x_1) \oplus f_2(x_2) \oplus ... \oplus f_n(x_n)$. Thus our characterization implies that if a Boolean function f is $\left\lceil \frac{n}{2} \right\rceil$ – private, then it is also n – private. Interestingly, the same characterization remains valid under a weaker definition of privacy. Specifically, this weaker definition allows

[1]The information theoretic approach alleviates the need for restricting the computational power of the participants, as well as the use of unproven intractability assumptions. The case of computationally bounded participants is entirely different, and is handled in [Y4,GMW].

coalitions to infer a limited (quite substantial) amount of additional information, and there can be a positive probability of error in computing the final output of the protocol. Finally, combining our result with [BGW], we conclude that there is a surprising gap in the Boolean privacy hierarchy: Every Boolean function, defined over a finite domain, is either exactly $\left\lfloor \frac{n-1}{2} \right\rfloor$ – private or exactly n – private, and there is nothing in between.

The rest of this paper is organized as follows: In section 2 we present the model and the definitions of privacy. In section 3 we consider the two-party case. Section 4 contains our main results: the characterization for the multi-party case, as well as some implications and conclusions.

2 Model and Definitions

In this section we define the model of distributed computation that is used in the sequel. We then proceed to the formal definitions of strong and weak privacy in this model.

The system consists of a complete synchronous network of n honest parties $P_1, P_2, ..., P_n$ with secure and reliable point-to-point communication (no eavesdropping). (By saying that the parties are honest it is meant that they send messages according to the protocol F.) At the beginning of an execution, each party P_i has an input x_i taken from a non-empty set of possible inputs A_i (no probability space is associated with A_i). In addition, each party has a random input r_i taken from a source of randomness R_i. The parties wish to compute a Boolean function $f : A_1 \times A_2 \times \ldots \times A_n \to \{0,1\}$. To this end, they exchange messages as prescribed by the protocol F. Messages are sent in rounds, where in each round every processor sends a message to every other processor. Each message a party sends in the $k - th$ round is determined using its input, its random input, the messages it received so far, and the identity of the receiver. As commonly assumed, the messages sent at each round are prefix free. We say that a protocol F computes the function f if the last message in the protocol, $F(x_1, \ldots, x_n)$, is an identical message sent by party P_1 to all parties, which contains the value $f(x_1, \ldots, x_n)$.

The *communication* passed in the network when the parties have inputs $\vec{x}$ and random inputs $\vec{r}$ is denoted $S(\vec{x}, \vec{r})$. Formally, the communication S is an n-by-n matrix whose (i, j) entry is the concatenation of all messages sent from P_i to P_j. For any $T \subset \{1, 2, \ldots, n\}$, S_T denotes the matrix S where entries (i, j) with either $i, j \in T$ or $i, j \notin T$ are omitted. That is, S_T is the communication between processors in T and in $\bar{T}$.

We say that a coalition (i.e. a set of parties) T *does not learn any additional information* (other than what follows from its input and the function value) from the execution of a randomized protocol F, which computes f, if the following holds: For every two inputs $\vec{x}, \vec{y} \in A_1 \times A_2 \times \ldots \times A_n$ that agree in their T entries (i.e. $\forall i \in T : x_i = y_i$) and satisfy $f(\vec{x}) = f(\vec{y})$, and for every choice of random inputs $\{r_i\}_{i \in T}$, the messages passed between T and $\bar{T}$ are identically distributed. That is:

$$< S_T(\{x_i\}_{i\in T}, \{r_i\}_{i\in T}, \{x_i\}_{i\in \bar{T}}) >=< S_T(\{y_i\}_{i\in T}, \{r_i\}_{i\in T}, \{y_i\}_{i\in \bar{T}}) >$$

where the probability space is over all random inputs in $\bar{T}$, namely $\{r_i\}_{i\in\bar{T}}$ (each r_i is distributed according to R_i and are all independent).

We say that a protocol F for computing f is *(strongly) t-private* if any coalition T of size $\leq t$ does not learn any additional information from the execution of the protocol. We say that a function f is *(strongly) t-private* if there exists a (strongly) t-private protocol that computes it.

The weak notion of privacy is different from the strong one, described above, in two ways: (a) coalitions may get some (limited) additional information (other than what follows from the inputs of the coalition members and the function value), and (b) the protocol may not always compute the correct value of the function. This is formalized as follows:

(a) Given $0 \leq \delta \leq 1$, we say that a protocol F for computing f is $(\delta, t) - private$ if the following holds: Let T be any coalition of size $\leq t$, and let $\vec{x}, \vec{y} \in A_1 \times A_2 \times \ldots \times A_n$ be any two inputs that agree in their T entries and satisfy $f(\vec{x}) = f(\vec{y})$. Then the *variation distance* on the space $\mathcal{S}_T$ (messages passed between T and $\bar{T}$), given $\vec{x}$ and given $\vec{y}$, is bounded above by δ. That is,

$$\frac{1}{2}\sum_{s\in\mathcal{S}_T} |Pr(s|\vec{x}) - Pr(s|\vec{y})| \leq \delta$$

(b) Given $0 \leq \varepsilon < 1$, we say that a protocol F has $\varepsilon - error$ in computing f if

$$\forall\vec{x} : Pr(F(\vec{x}) = f(\vec{x})) \geq 1 - \varepsilon \ ,$$

(in both (a) and (b) the probabilities are taken over the random inputs of all the participants).

We remark that even for $\delta = 0$, error free (δ, t) – privacy is a weaker requirement than (strong) t-privacy. In the special case of $n = 2$ we say that a function is (strongly) *private* if it is (strongly) 1 – private, and it is δ – private if it is $(\delta, 1)$ – private.

3 The Two-Party Case

In this section we consider the case where f is a Boolean function of two variables. We show that if f is weakly private then it can be expressed as $f(x_1, x_2) = f_1(x_1) \oplus f_2(x_2)$, where f_1 and f_2 are also Boolean functions. On the other hand we show that functions of the form $f(x_1, x_2) = f_1(x_1) \oplus f_2(x_2)$, can be computed in a strongly private way.

We now present a lemma that will play a central rule in the proof of the characterization theorem.

Lemma 1*:* *Let $\varepsilon, \delta \geq 0$ satisfy $\varepsilon + \delta < \frac{1}{2}$. Let A_1, A_2, B be non-empty sets and let $f : A_1 \times A_2 \rightarrow B$ be a function that can be computed δ – privately with ε – error. Under these assumptions, for every $b \in B, x_1, y_1 \in A_1$, and $x_2, y_2 \in A_2$ the following condition holds: If $f(x_1, x_2) = f(x_1, y_2) = f(y_1, x_2) = b$, then $f(y_1, y_2) = b$.*

Before proving Lemma 1 we introduce three technical lemmas. The first one holds for any two-party communication protocol. It relates the probabilities of any communication string s to be sent on the four corners of any "input rectangle" (x_1, x_2), (x_1, y_2), (y_1, x_2), (y_1, y_2). (An equivalent lemma is implicitly used by Paturi and Simon [PS]).

Lemma 2: *Let A_1, A_2 be non-empty sets, and F a two-party communication protocol. For every communication string s and all inputs $x_1, y_1 \in A_1, x_2, y_2 \in A_2$*

$$Pr(s|(x_1, x_2)) \cdot Pr(s|(y_1, y_2)) = Pr(s|(x_1, y_2)) \cdot Pr(s|(y_1, x_2))$$

Proof: Let $s = m_1 \circ m_2 \circ ... \circ m_k$, where m_i is the message sent in the i-th round, and $\circ$ denotes concatenation. Assume, without loss of generality, that the first message of the protocol is sent by P_1 and that k (the number of rounds) is even. Let $Pr_1(s|x_1)$ denote the probability that P_1 will send the messages $m_1, m_3, m_5...$ given that its input is x_1, and that the messages received from P_2 are $m_2, m_4, m_6....$ That is,

$$Pr_1(s|x_1) = Pr(m_1|x_1) \cdot Pr(m_3|x_1, m_1, m_2) \cdot \ldots \cdot Pr(m_{k-1}|x_1, m_1, m_2, ..., m_{k-2})$$

and similarly $Pr_2(s|x_2)$ is defined as

$$Pr_2(s|x_2) = Pr(m_2|x_2, m_1) \cdot Pr(m_4|x_2, m_1, m_2, m_3) \cdot \ldots \cdot Pr(m_k|x_2, m_1, m_2, ..., m_{k-1}).$$

(Recall that every message is a function of the local input, the messages received from the other party, and the random input.) By these definitions, for every input (x_1, x_2), and every communication string s we have:

$$Pr(s|(x_1, x_2)) \;=\; Pr_1(s|x_1) \;\cdot\; Pr_2(s|x_2)$$

and therefore

$$\begin{aligned} Pr(s|(x_1, x_2)) \cdot Pr(s|(y_1, y_2)) &= Pr_1(s|x_1) \cdot Pr_2(s|x_2) \cdot Pr_1(s|y_1) \cdot Pr_2(s|y_2) \\ &= Pr(s|(x_1, y_2)) \cdot Pr(s|(y_1, x_2)) \; . \end{aligned}$$

This completes the proof of the lemma. □

The second technical lemma gives a lower bound on the probability of any communication string s to be sent on the input (y_1, y_2), given the probabilities that s will be sent on each of the inputs $(x_1, x_2), (x_1, y_2)$, and (y_1, x_2).

Lemma 3: *Let $0 \leq p_1, p_2, p_3, p_4 \leq 1$ such that $p_1 \cdot p_4 = p_2 \cdot p_3$:*
(1) If $p_1 \leq p_2, p_3$ then $p_4 \geq p_1$.
(2) If $p_1 \geq p_2, p_3$ then $p_4 \geq p_1 - (p_1 - p_2) - (p_1 - p_3)$.
(3) If $p_2 \leq p_1 \leq p_3$ then $p_4 \geq p_1 - (p_1 - p_2)$.
(4) If $p_3 \leq p_1 \leq p_2$ then $p_4 \geq p_1 - (p_1 - p_3)$.

Proof: We prove each of the four cases using simple arithmetic manipulations.

(1) In case that $p_1 \leq p_2, p_3$ if $p_1 = 0$ then clearly $p_4 \geq p_1$. Otherwise, the following holds

$$p_4 = \frac{p_2 \cdot p_3}{p_1} \geq \frac{p_1 \cdot p_1}{p_1} = p_1$$

(2) In case that $p_1 \geq p_2, p_3$, if $p_1 = 0$ then $p_2 = p_3 = 0$, and thus the inequality holds. Otherwise we have

$$\begin{aligned} p_4 &= \frac{p_2 \cdot p_3}{p_1} \\ &= \frac{(p_1 - (p_1 - p_2)) \cdot (p_1 - (p_1 - p_3))}{p_1} \\ &= \frac{p_1^2 - (p_1 - p_2) \cdot p_1 - (p_1 - p_3) \cdot p_1 + ((p_1 - p_2) \cdot (p_1 - p_3))}{p_1} \\ &\geq p_1 - (p_1 - p_2) - (p_1 - p_3) \end{aligned}$$

(3) In case that $p_2 \leq p_1 \leq p_3$ we have to show that $p_4 \geq p_2$. If $p_3 = 0$ then so is p_2 and the inequality trivially holds. Otherwise, assume by way of contradiction that $p_4 < p_2$. Since $p_1 \leq p_3$ then $p_1 \cdot p_4 < p_3 \cdot p_2$, contradicting $p_1 \cdot p_4 = p_2 \cdot p_3$.

(4) Similar to the proof of (3).

This completes the proof of the lemma. □

The next lemma is a trivial property of the variation distance.

Lemma 4: *Let S be the set of all communication strings and let p_1 and p_2 be two probability distributions defined over S. Denote by $S_1 \subseteq S$ the set $\{s | p_1(s) \geq p_2(s)\}$, then*

$$\sum_{s \in S_1} (p_1(s) - p_2(s)) = \frac{1}{2} \sum_{s \in S} |p_1(s) - p_2(s)|$$

Proof: The proof is by simple arithmetic manipulations:

$$\begin{aligned} 2 \cdot \sum_{s \in S_1} (p_1(s) - p_2(s)) &= 2 \cdot \sum_{s \in S_1} p_1(s) - 2 \cdot \sum_{s \in S_1} p_2(s) \\ &= \sum_{s \in S_1} p_1(s) + \left(1 - \sum_{s \in \bar{S}_1} p_1(s)\right) - \sum_{s \in S_1} p_2(s) - \left(1 - \sum_{s \in \bar{S}_1} p_2(s)\right) \\ &= \sum_{s \in S_1} (p_1(s) - p_2(s)) - \sum_{s \in \bar{S}_1} (p_1(s) - p_2(s)) \\ &= \sum_{s \in S} |p_1(s) - p_2(s)| \end{aligned}$$

(The last equality follows from the definition of S_1.) □

Using these three lemmas, we can now prove Lemma 1.

Proof: (of Lemma 1).

Let F be a protocol that computes f δ – privately with ε – error. Let S_b be the set of all communication strings whose last message is b. Recall that on the first three points, these strings correspond to executions computing the correct value of the function (which equals b). Define

$$\begin{aligned}
S_b^1 &\triangleq \{s|s \in S_b \text{ and } Pr(s|(x_1,x_2)) \le Pr(s|(x_1,y_2)), Pr(s|(y_1,x_2))\} \\
S_b^2 &\triangleq \{s|s \in S_b \text{ and } Pr(s|(x_1,x_2)) \ge Pr(s|(x_1,y_2)), Pr(s|(y_1,x_2))\} \\
S_b^3 &\triangleq \{s|s \in S_b \text{ and } Pr(s|(x_1,y_2)) < Pr(s|(x_1,x_2)) < Pr(s|(y_1,x_2))\} \\
S_b^4 &\triangleq \{s|s \in S_b \text{ and } Pr(s|(y_1,x_2)) < Pr(s|(x_1,x_2)) < Pr(s|(x_1,y_2))\}
\end{aligned}$$

The protocol F has at most ε – error for every input. Thus, to prove that $f(y_1,y_2) = b$, it suffices to show that on input (y_1,y_2), the probability of having a communication string whose last message is b (that is $s \in S_b$), is greater than ε:

$$\begin{aligned}
\sum_{s\in S_b} Pr(s|(y_1,y_2)) &= \sum_{s\in S_b^1} Pr(s|(y_1,y_2)) + \sum_{s\in S_b^2} Pr(s|(y_1,y_2)) \\
&+ \sum_{s\in S_b^3} Pr(s|(y_1,y_2)) + \sum_{s\in S_b^4} Pr(s|(y_1,y_2))
\end{aligned}$$

Now, for each set S_b^i we use the appropriate part of Lemma 3, together with Lemma 2. These imply that this sum is bounded below by

$$\begin{aligned}
& \sum_{s\in S_b^1} Pr(s|(x_1,x_2)) \\
+ & \sum_{s\in S_b^2} Pr(s|(x_1,x_2)) - \sum_{s\in S_b^2} (Pr(s|(x_1,x_2)) - Pr(s|(x_1,y_2))) - \sum_{s\in S_b^2} (Pr(s|(x_1,x_2)) - Pr(s|(y_1,x_2))) \\
+ & \sum_{s\in S_b^3} Pr(s|(x_1,x_2)) - \sum_{s\in S_b^3} (Pr(s|(x_1,x_2)) - Pr(s|(x_1,y_2))) \\
+ & \sum_{s\in S_b^4} Pr(s|(x_1,x_2)) - \sum_{s\in S_b^4} (Pr(s|(x_1,x_2)) - Pr(s|(y_1,x_2))) \,.
\end{aligned}$$

This last expression equals

$$\begin{aligned}
\sum_{s\in S_b} Pr\left(s|\left(x_1,x_2\right)\right) &- \sum_{s\in S_b^2\cup S_b^3} \left(Pr\left(s|\left(x_1,x_2\right)\right) - Pr\left(s|\left(x_1,y_2\right)\right)\right) \\
&- \sum_{s\in S_b^2\cup S_b^4} \left(Pr\left(s|\left(x_1,x_2\right)\right) - Pr\left(s|\left(y_1,x_2\right)\right)\right)
\end{aligned}$$

The first summand is at least $1-\varepsilon$ since ε is the maximum error permitted on the input (x_1,x_2). According to Lemma 4 and the fact that the protocol F is δ – private, each of the other two summands is at most δ. Thus we have

$$\sum_{s \in S_b} Pr(s|(y_1, y_2)) \geq 1 - \varepsilon - \delta - \delta \ .$$

Since $\varepsilon + \delta < \frac{1}{2}$, we have $1 - \varepsilon - \delta - \delta > \varepsilon$. This completes the proof of Lemma 1. $\square$

Lemma 1 suffices for showing that certain Boolean functions of two variables are not weakly private. The first example, which was given in [BGW] (with respect to strong privacy), is the OR function ($A_1 = A_2 = \{0,1\}$ and $f(x_1, x_2) = x_1 \vee x_2$). Clearly this function does not satisfy Lemma 1. For an additional example we take A_1 and A_2 to be the set of all integers and f the IDENTITY function ($f(x_1, x_2) = 1 \Leftrightarrow x_1 = x_2$). This function is not private since for any c we have $f(c-1, c) = f(c-1, c+1) = f(c, c+1) = 0$ but $f(c, c) = 1$.

Theorem 1: *Let $\varepsilon, \delta \geq 0$ satisfy $\varepsilon + \delta < \frac{1}{2}$. Let A_1, A_2 be non-empty sets and $f : A_1 \times A_2 \to \{0,1\}$ an arbitrary Boolean function. Then f can be computed δ – privately with ε – error if and only if there exist Boolean functions $f_1 : A_1 \to \{0,1\}$, $f_2 : A_2 \to \{0,1\}$ such that $f(x_1, x_2) = f_1(x_1) \oplus f_2(x_2)$.*

Proof: First, we present a private protocol for computing any function f of the form $f(x_1, x_2) = f_1(x_1) \oplus f_2(x_2)$:

(1) P_1 computes $f_1(x_1)$ and sends its value (one bit) to P_2.

(2) P_2 computes $f_2(x_2)$ and sends $f(x_1, x_2) = f_1(x_1) \oplus f_2(x_2)$ (one bit) to P_1.

It is clear that P_1 does not learn any additional information since the only message it received during the protocol contains the function value, and P_2 does not learn any additional information since it can compute by itself (from the function value and its input) $f_1(x_1) = f(x_1, x_2) \oplus f_2(x_2)$. The above protocol computes f with no errors and with strong privacy. In addition, this protocol is deterministic and the number of bits exchanged is exactly 2.

Now, we assume that f can be computed δ – privately with ε – error, and we show how to find f_1 and f_2 as stated. Let α be an arbitrary element of A_2, and define the following sets:

$$B \stackrel{\triangle}{=} \{x_1 \in A_1 | f(x_1, \alpha) = 0\}$$

$$C \stackrel{\triangle}{=} \{x_2 \in A_2 | \forall x_1 \in B : f(x_1, x_2) = 0\}$$

We assume, without loss of generality, that there exists some x_1 such that $f(x_1, \alpha) = 0$ (that is, B is not empty). We will show now that the function is constant over each of the domains $B \times C$, $B \times \bar{C}$, $\bar{B} \times C$, $\bar{B} \times \bar{C}$:

Claim 1: $\forall x_1 \in B \, \forall x_2 \in C : f(x_1, x_2) = 0$.

Proof: This follows directly from the definition of C.

Claim 2: $\forall x_1 \in B \, \forall x_2 \in \bar{C} : f(x_1, x_2) = 1$.

Proof: Assume to the contrary that there exist $x_1 \in B$ and $x_2 \in \bar{C}$ such that $f(x_1, x_2) = 0$. Now, by the definition of C, $x_2 \in \bar{C} \Rightarrow \exists y_1 \in B$ such that $f(y_1, x_2) = 1$. By the definition of B, $x_1, y_1 \in B \Rightarrow f(x_1, \alpha) = f(y_1, \alpha) = 0$. Since we have $f(x_1, \alpha) = f(y_1, \alpha) = f(x_1, x_2) = 0$, then according to Lemma 1 we must also have $f(y_1, x_2) = 0$ – contradiction.

Claim 3*:* $\forall x_1 \in \bar{B}\, \forall x_2 \in C : f(x_1, x_2) = 1$.

Proof: Assume to the contrary that there exist $x_1 \in \bar{B}$ and $x_2 \in C$ such that $f(x_1, x_2) = 0$. It follows from the definition of $\bar{B}$ that $x_2 \neq \alpha$. Let y_1 be an arbitrary element of B (recall that B is not empty). Now, $y_1 \in B \Rightarrow f(y_1, \alpha) = 0$ and $x_2 \in C \Rightarrow f(y_1, x_2) = 0$. Since we have $f(y_1, \alpha) = f(y_1, x_2) = f(x_1, x_2) = 0$ then according to Lemma 1 we must also have $f(x_1, \alpha) = 0$ – contradicting the fact that $x_1 \in \bar{B}$.

Claim 4*:* $\forall x_1 \in \bar{B}\, \forall x_2 \in \bar{C} : f(x_1, x_2) = 0$.

Proof: Assume to the contrary that there exist $x_1 \in \bar{B}$ and $x_2 \in \bar{C}$ such that $f(x_1, x_2) = 1$. Recall that $x_1 \in \bar{B}$ implies $f(x_1, \alpha) = 1$ and let y_1 be an arbitrary element of B i.e $f(y_1, \alpha) = 0$. According to claim 2 $f(y_1, x_2) = 1$. Since we have $f(x_1, x_2) = f(y_1, x_2) = f(x_1, \alpha) = 1$ then according to Lemma 1 we must also have $f(y_1, \alpha) = 1$ – contradicting the fact that $y_1 \in B$.
We now define:

$$f_1(x_1) = \begin{cases} 0 & \text{if } x_1 \in B \\ 1 & \text{if } x_1 \notin B \end{cases}$$

$$f_2(x_2) = \begin{cases} 0 & \text{if } x_2 \in C \\ 1 & \text{if } x_2 \notin C \end{cases}$$

then by claims 1-4 we have $f(x_1, x_2) = f_1(x_1) \oplus f_2(x_2)$ for each of the four possible combinations $(x_1, x_2) \in B \times C$, $B \times \bar{C}$, $\bar{B} \times C, \bar{B} \times \bar{C}$.
This completes the proof of theorem 1. □

One conclusion of theorem 1 is that in the two-party case, if f can be privately computed then it can be privately computed by a deterministic protocol. We emphasize that this does not hold for the multi-party case. A second conclusion is that in the two-party case whatever can be privately computed under the weak notion can also be privately computed under the strong notion. Thus (for any $\varepsilon, \delta \geq 0$ such that $\varepsilon + \delta < \frac{1}{2}$) these two notions are equivalent. As we will see in the next section, this conclusion holds in the multi-party case as well.

4 The Multi Party Case

In this section we prove the main result of our paper: A complete characterization of $n-$ variable Boolean functions that are $\left\lceil \frac{n}{2} \right\rceil -$ private. We start with a lemma that helps reduce the multi party case to the two party scenario. Using this lemma, we proceed to a detailed proof of the characterization theorem. Finally, we give some implications and corollaries. Throughout this section, we will say that a function f is *weakly t–private* if there are $\varepsilon, \delta \geq 0$ satisfying $\varepsilon + \delta < \frac{1}{2}$, such that f can be computed (δ, t)–privately with ε–error.

Lemma 5: *Let $A_1, A_2, \ldots, A_n$ be non-empty sets, $\varepsilon, \delta \geq 0$ satisfying $\varepsilon + \delta < \frac{1}{2}$, and $f : A_1 \times A_2 \times \ldots \times A_n \to \{0,1\}$ be $\left(\delta, \left\lceil \frac{n}{2} \right\rceil\right)$ – privately computable with ε– error. Let $S \subseteq \{1, 2, \ldots, n\}$ be any subset of size $\left\lceil \frac{n}{2} \right\rceil$. Denote by D (resp. E) the Cartesian product of the A_i with $i \in S$ (resp. $i \in \bar{S}$). Then, viewing f as a two argument function $f : D \times E \to \{0,1\}$, f is δ–private with ε– error.*

Proof: Given an n-party protocol for computing $f : A_1 \times A_2 \times \ldots \times A_n \to \{0,1\}$ $\left(\delta, \left\lceil \frac{n}{2} \right\rceil\right)$ – privately with ε–error, we convert it into a two party protocol for computing $f : D \times E \to \{0,1\}$. Denote the two parties by Q_1 and Q_2. Q_1 simulates the role of the $\left\lceil \frac{n}{2} \right\rceil$ processors P_i with $i \in S$, using its source of random bits as $\left\lceil \frac{n}{2} \right\rceil$ independent sources of random bits. (Q_2 acts similarly with respect to $\bar{S}$ whose size is $\left\lfloor \frac{n}{2} \right\rfloor$.) The messages exchanged between Q_1 and Q_2 in this two party protocol correspond to messages exchanged between S and $\bar{S}$ processors in the original multi party protocol. Using the definitions, it is easy to see that this two party protocol computes $f : D \times E \to \{0,1\}$ δ–privately with ε–error. $\square$

We remark that to make use of the $\left\lceil \frac{n}{2} \right\rceil$–privacy, both S and $\bar{S}$ must be of size not exceeding $\left\lceil \frac{n}{2} \right\rceil$. Our main theorem states that if $f : A_1 \times A_2 \times \ldots \times A_n \to \{0,1\}$ is weakly $\left\lceil \frac{n}{2} \right\rceil$ – private, then f can be expressed as the exclusive-or of n Boolean functions $f_1, f_2, \ldots, f_n$. The proof makes use of theorem 1 and of lemma 5.

Theorem 2: *Let $A_1, A_2, \ldots, A_n$ be non-empty sets, and $f : A_1 \times A_2 \times \ldots \times A_n \to \{0,1\}$. Suppose f is weakly $\left\lceil \frac{n}{2} \right\rceil$ – private. Then there are n Boolean functions $f_1 : A_1 \to \{0,1\}$, $f_2 : A_2 \to \{0,1\}$, $\ldots$, $f_n : A_n \to \{0,1\}$ such that*

$$f(x_1, x_2, \ldots, x_n) = f_1(x_1) \oplus f_2(x_2) \oplus \ldots \oplus f_n(x_n) \; .$$

Proof: The proof consists of two parts. In the first part we show that for every i, the set A_i can be partitioned into two disjoint sets

$$A_i = B_i \bigcup C_i \quad (B_i \bigcap C_i = \emptyset, \; B_i \neq \emptyset)$$

such that for all $b_i \in B_i, \check{b}_i \in B_i, c_i \in C_i, x_j \in A_j$ $(j \neq i)$

$$f(x_1, \ldots, x_{i-1}, b_i, x_{i+1}, \ldots, x_n) \neq f(x_1, \ldots, x_{i-1}, c_i, x_{i+1}, \ldots, x_n) \tag{1}$$

and

$$f(x_1, \ldots, x_{i-1}, b_i, x_{i+1}, \ldots, x_n) = f(x_1, \ldots, x_{i-1}, \check{b}_i, x_{i+1}, \ldots, x_n) \tag{2}$$

In the second part of the proof, we show how to derive the desired characterization of f from this property.

To simplify the exposition, the first part is proven for $i = 1$, and the subscript is omitted from the two sets in the partition. We use the following notations

$$\begin{aligned} R &\triangleq \left\{2, 3, \ldots, \left\lceil \frac{n}{2} \right\rceil \right\} \\ T &\triangleq \left\{ \left\lceil \frac{n}{2} \right\rceil + 1, \ldots, n-1 \right\} \\ S_1 &\triangleq \{1\} \bigcup R \\ S_2 &\triangleq \{n\} \bigcup R \ . \end{aligned}$$

Each of the sets S_1 and S_2 are of size $\left\lceil \frac{n}{2} \right\rceil$, and thus Lemma 5 applies to both. We establish the existence of the desired partition by examining the effect of "switching" the variable x_1 – from S_1 to $\bar{S_2}$. By Lemma 5 together with theorem 1, there are functions

$$g : \times_{i \in S_1} A_i \to \{0, 1\}$$

$$h : \times_{i \in \bar{S_1}} A_i \to \{0, 1\}$$

such that for every $x_i \in A_i$

$$f(x_1, \ldots, x_n) = g\left(x_1, \ldots, x_{\lceil \frac{n}{2} \rceil}\right) \oplus\ h\left(x_{\lceil \frac{n}{2} \rceil + 1}, \ldots, x_n\right) \tag{3}$$

Similarly, there are functions

$$\begin{aligned} \tilde{g} &: \times_{i \in S_2} A_i \to \{0, 1\} \\ \tilde{h} &: \times_{i \in \bar{S_2}} A_i \to \{0, 1\} \end{aligned}$$

such that

$$f(x_1, \ldots, x_n) = \tilde{g}\left(x_n, x_2 \ldots, x_{\lceil \frac{n}{2} \rceil}\right) \oplus\ \tilde{h}\left(x_{\lceil \frac{n}{2} \rceil + 1}, \ldots, x_{n-1}, x_1\right) \tag{4}$$

We distinguish between two cases. If f does not depend on x_1, its first argument, then we simply take $B = A_1$ and $C = \emptyset$. The interesting case is where f does depend on x_1. That is, there are $b, c \in A_1$ $(b \neq c)$, $\vec{\alpha} \in \times_{i \in R} A_i$, $\vec{\beta} \in \times_{i \in T} A_i$, $d \in A_n$ such that

$$f(b, \vec{\alpha}, \vec{\beta}, d) \neq f(c, \vec{\alpha}, \vec{\beta}, d) \ .$$

Define the sets

$$B \triangleq \left\{ a_1 \in A_1 \,|\, f(a_1, \vec{\alpha}, \vec{\beta}, d) = f(b, \vec{\alpha}, \vec{\beta}, d) \right\} \ ,$$

$$C \triangleq \left\{ a_1 \in A_1 \,|\, f(a_1, \vec{\alpha}, \vec{\beta}, d) = f(c, \vec{\alpha}, \vec{\beta}, d) \right\} \ .$$

Since f is a Boolean function, $A_1 = B \bigcup C$. By the definition, for all $b \in B$, $c \in C$

$$f\left(b, \vec{\alpha}, \vec{\beta}, d\right) \neq f\left(c, \vec{\alpha}, \vec{\beta}, d\right) \tag{5}$$

Assume, by way of contradiction, the existence of $b \in B, c \in C, \vec{x}_R \in \times_{i \in R} A_i, \vec{x}_T \in \times_{i \in T} A_i, x_n \in A_n$ such that

$$f(b, \vec{x}_R, \vec{x}_T, x_n) = f(c, \vec{x}_R, \vec{x}_T, x_n) \ .$$

By (3), we have

$$g(b, \vec{x}_R) \oplus h(\vec{x}_T, x_n) = g(c, \vec{x}_R) \oplus h(\vec{x}_T, x_n)$$

and thus

$$g(b, \vec{x}_R) = g(c, \vec{x}_R) \ .$$

Using (3) again, this implies

$$f(b, \vec{x}_R, \vec{\beta}, d) = f(c, \vec{x}_R, \vec{\beta}, d) \ .$$

Now, using (4), this implies

$$\tilde{h}(\vec{\beta}, b) = \tilde{h}(\vec{\beta}, c)$$

and thus

$$\begin{aligned} f(b, \vec{\alpha}, \vec{\beta}, d) &= \tilde{g}(d, \vec{\alpha}) \oplus \tilde{h}(\vec{\beta}, b) \\ &= \tilde{g}(d, \vec{\alpha}) \oplus \tilde{h}(\vec{\beta}, c) \\ &= f(c, \vec{\alpha}, \vec{\beta}, d) \end{aligned}$$

contradicting (5).
Thus for every $b \in B, \check{b} \in B, c \in C, \vec{x}_R \in \times_{i \in R} A_i, \vec{x}_T \in \times_{i \in T} A_i, x_n \in A_n$

$$f(b, \vec{x}_R, \vec{x}_T, x_n) \neq f(c, \vec{x}_R, \vec{x}_T, x_n)$$

$$f(\check{b}, \vec{x}_R, \vec{x}_T, x_n) \neq f(c, \vec{x}_R, \vec{x}_T, x_n) \ .$$

Again, since f is Boolean, these two inequalities imply

$$f(b, \vec{x}_R, \vec{x}_T, x_n) = f(\check{b}, \vec{x}_R, \vec{x}_T, x_n)$$

which completes the first part of the proof.

In the second part of the proof, we show that if for every i the set A_i can be partitioned such that 1 and 2 hold, then the function f has the desired form. We begin the second part of the proof by fixing an element $\tilde{b}_i \in B_i$ for each $i = 1, 2, \ldots n$. Without loss of generality, assume

$$f(\tilde{b}_1, \tilde{b}_2, \ldots, \tilde{b}_n) = 0 \ .$$

Define the functions $f_i : A_i \rightarrow \{0,1\}$ by

$$f_i(x_i) = \begin{cases} 0 & \text{if } x_i \in B_i \\ 1 & \text{if } x_i \in C_i \end{cases}$$

Given any $x_1 \in A_1$, …, $x_n \in A_n$, let $J \subseteq \{1,\ldots,n\}$ be the set of indices of the x_i's in C_i, and $M \subseteq \{1,\ldots,n\}$ its complement. Denote by ℓ the size of J, and let $k = n - \ell$. We will index the elements in J and M separately, that is

$$(x_1,\ldots,x_n) = (x_{j_1}, x_{j_2}, \ldots, x_{m_1}, \ldots, x_{m_k}, \ldots, x_{j_\ell}) \ .$$

By (1) and (2)

$$\begin{aligned}
f(x_1,\ldots,x_n) &= f(x_{j_1}, x_{j_2}, \ldots, x_{m_1}, \ldots, x_{m_k}, \ldots, x_{j_\ell}) \\
&= f(x_{j_1}, x_{j_2}, \ldots, \tilde{b}_{m_1}, \ldots, \tilde{b}_{m_k}, \ldots, x_{j_\ell}) \\
&= f(\tilde{b}_{j_1}, x_{j_2}, \ldots, \tilde{b}_{m_1}, \ldots, \tilde{b}_{m_k}, \ldots, x_{j_\ell}) \oplus 1 \\
&= f(\tilde{b}_{j_1}, \tilde{b}_{j_2}, \ldots, \tilde{b}_{m_1}, \ldots, \tilde{b}_{m_k}, \ldots, x_{j_\ell}) \oplus 1 \oplus 1 \\
&\ \vdots \\
&= f(\tilde{b}_{j_1}, \tilde{b}_{j_2}, \ldots, \tilde{b}_{m_1}, \ldots, \tilde{b}_{m_k}, \ldots, \tilde{b}_{j_\ell}) \oplus (\ell \bmod 2) \\
&= \ell \pmod 2 \ .
\end{aligned}$$

By the definition of the f_i's

$$f_1(x_1) \oplus f_2(x_2) \oplus \ldots \oplus f_n(x_n) = \ell \pmod 2$$

and thus

$$f(x_1, x_2, \ldots, x_n) = f_1(x_1) \oplus f_2(x_2) \oplus \ldots \oplus f_n(x_n) \ .$$

for each $\vec{x} \in \times_{i=1}^{n} A_i$. □

We now turn to some implications of theorem 2. First we note that if $f : A_1 \times A_2 \times \ldots \times A_n \to \{0,1\}$ has the form

$$f(x_1, x_2, \ldots, x_n) = f_1(x_1) \oplus f_2(x_2) \oplus \ldots \oplus f_n(x_n)$$

then there is a very simple protocol [Bh] for computing f n – privately. The i-th participant locally computes the bit $y_i \triangleq f_i(x_i)$. Then, it picks $n-1$ random independent bits $y_{i,1}, y_{i,2}, \ldots, y_{i,n-1}$, and $y_{i,n}$ such that $y_i = y_{i,1} \oplus y_{i,2} \oplus \ldots \oplus y_{i,n}$ holds. It sends $y_{i,j}$ to the j-th participant over their joint channel. After getting the n splits $y_{1,i}, y_{2,i}, \ldots, y_{n,i}$, the i-th participant adds them modulo 2 and sends the result $z_i \triangleq y_{1,i} \oplus y_{2,i} \oplus \ldots \oplus y_{n,i}$ to every other participant. The sum modulo 2 of these n z_i's equals $f(x_1, x_2, \ldots, x_n)$. This protocol is (strongly) t-private for any $1 \le t \le n$. Thus we have

Theorem 3*:* *Let $f : A_1 \times A_2 \times \ldots \times A_n \to \{0,1\}$. If f is weakly $\left\lceil \frac{n}{2} \right\rceil$ – private, then it is (strongly) $n-$ private.*

In particular, there is no Boolean function that is t – private but not $t+1$ – private for any t in the range $\left\lceil \frac{n}{2} \right\rceil \le t < n$.

We now consider the case where f is a Boolean function of Boolean variables. There are only four Boolean functions of a Boolean variable (the two constants, the variable, and its complement). The form of private Boolean functions of Boolean variables is thus particularly simple.

Theorem 4: *A function $f : \{0,1\}^n \rightarrow \{0,1\}$ is n-private if and only if there is a subset $J \subseteq \{1, \ldots, n\}$ such that*

$$\begin{aligned} \textit{either} \quad f(x_1, x_2, \ldots, x_n) &= \bigoplus_{j \in J} x_j \\ \textit{or} \quad \bar{f}(x_1, x_2, \ldots, x_n) &= \bigoplus_{j \in J} x_j \ . \end{aligned}$$

Finally, we remark that our characterization of Boolean functions that are $\left\lceil \frac{n}{2} \right\rceil$ – private is valid even for functions defined over infinite domains. The $\left\lfloor \frac{n-1}{2} \right\rfloor$ – private protocol of [BGW], on the other hand, heavily relied on the finiteness of the domains. We believe that for infinite domains, there exist functions that are not $\left\lfloor \frac{n-1}{2} \right\rfloor$ – private. Indeed, the secret-sharing techniques used in that $\left\lfloor \frac{n-1}{2} \right\rfloor$ – private protocol, cannot be utilized in countable domains, as shown in [CKss].

5 Acknowledgements

We would like to thank Shai Ben-David and Oded Goldreich for helpful discussions and comments on the topics of this paper.

6 References

[Bh] Benaloh (Cohen), J.D., "Secret Sharing Homomorphisms: Keeping Shares of a Secret Secret", *Advances in Cryptography - Crypto86 (proceedings)*, A.M. Odlyzko (ed.), Springer-Verlag, Lecture Notes in Computer Science, Vol. 263, pp. 251-260, 1987.

[BGW] Ben-or M., S. Goldwasser, and A. Wigderson, "Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation" *Proc. of 20th STOC*, 1988, pp. 1-10.

[CCD] Chaum, D., C. Crepeau, and I. Damgard, "Multiparty Unconditionally Secure Protocols" *Proc. of 20th STOC*, 1988, pp. 11-19.

[CKss] Chor, B., and E. Kushilevitz, "Secret Sharing Over Infinite Domains", *Advances in Cryptography - Crypto89 (proceedings)* - to appear.

[GMW] Goldreich, O., S. Micali, and A. Wigderson, "How to Play Any Mental Game", *Proc. of 19th STOC*, 1987, pp. 218-229.

[PS] Paturi, R., and J. Simon, "Probabilistic communication complexity", *Jour. of Computers and System Sciences*, Vol. 33, 1986, pp. 106-123.

[Y4] Yao, A. C., "How to Generate and Exchange Secrets", *Proc. of 27th FOCS*, 1986, pp. 162-167.