New Monotones and Lower Bounds in Unconditional Two-Party Computation

Stefan Wolf Jürg Wullschleger

***Abstract*— Since *oblivious transfer*, a primitive of paramount importance in secure two- and multi-party computation, cannot be realized in an unconditionally secure way for both parties from scratch, *reductions* to weak information-theoretic primitives as well as between different variants of the functionality are of great interest.**

In this context, various *monotones*—quantities that cannot be increased by any protocol—are introduced and then used to derive lower bounds on the *possibility* and *efficiency* of such reductions.

***Index Terms*— Lower bounds, monotones, oblivious transfer, two-party computation, unconditional security.**

I. Introduction

The advantage of *unconditional* or *information-theoretic* security—as compared to computational security—is that it does not depend on any assumption on an adversary's computing power or memory space, nor on the hardness of any computational problem. Its disadvantage, on the other hand, is that it cannot be realized from scratch. This is why *reductions* are of great interest and importance in this context: Which functionality can be realized from which other? If a reduction is possible in principle, what is the best efficiency, i.e., the minimum number of instances of the initial primitive required per realization of the target functionality?

A task of particular importance in secure two-party computation is *oblivious transfer*, which is known to be impossible to realize from scratch in an unconditionally secure way for both parties by any (classical or even quantum) protocol. On the other hand, it *can* be realized from noisy channels [7], [9], [12], weak versions of oblivious transfer [8], [3], [4], [13], [14], [28], or correlated pieces of information [25], [21].

For the same reason, reductions between different variants of oblivious transfer are of interest as well: chosen 1-out-of-2 oblivious transfer from Rabin oblivious transfer [6], *string* oblivious transfer from *bit* oblivious transfer [3], 1-out-of-n oblivious transfer from 1-out-of-2 oblivious transfer, oblivious transfer from A to B from oblivious transfer from B to A [10], [22], [27], and so forth. A number of lower bounds in the context of such reductions have been given, based on information-theoretic arguments [15], [19].

With respect to information-theoretic reductions between cryptographic and information-theoretic functionalities, quantities which never increase during the execution of a protocol—so-called *monotones* [5]—are of great importance. In *key agreement*, for instance, two parties A and B can start with correlated pieces of information X and Y, respectively, and try to generate a secret key S by public communication such that an adversary E, who initially knows a third random variable Z, is virtually ignorant about S. It has been shown in [23] that the *intrinsic information* [20] of A's and B's entire knowledge, given E's, is a monotone, i.e., cannot increase. This immediately leads to the following bound on the size of the generated key: $H(S) \leq I(X;Y{\downarrow}Z)$.

A. Contribution

In Section III, we define *monotones* as quantities that cannot be increased by any two-party computation. We explicitly give seven such monotones. In Section IV, we show how these monotones can be used to derive lower bounds on reductions between two-party primitives. In Section V, finally, we present concrete lower bounds.

II. Preliminaries

We say that two random variables X and Y are equivalent, denoted by $X \equiv Y$, if there exists a bijective function $g : \mathcal{X} \rightarrow \mathcal{Y}$ such that $Y = g(X)$ holds with probability 1.

Three random variables X, Y and Z form a *Markov-chain*, denoted by $X \leftrightarrow Y \leftrightarrow Z$, if $P_{Z|XY} = P_{Z|Y}$. This means that *X and Z are independent, given Y.*

A *(non-interactive) functionality P* takes as input the values x and y from the two players Alice and Bob, and returns them values u and v distributed accoring to a distribution $P_{UV|XY}$.

A *protocol* is a pair of functions (f,g) that is executed between Alice and Bob as follows. Let x and y be the inputs to the players, chosen according to a fixed distribution P_{XY}. The players choose uniformly at random $r_A, r_B \in \{0,1\}^*$ and then repeat for $i = 1, 2 \ldots$: If i is odd, then Alice sends a message $m_i = f(x, m_1, \ldots, m_{i-1}; r_A)$ to Bob; if i is even, Bob sends a message $m_i = g(y, m_1, \ldots, m_{i-1}; r_B)$ to Alice. If any m_i is equal to halt, i.e., if one of the two players aborts the computation, then the loop is exited. Finally, Alice outputs $u = f(x, m_1, \ldots, m_i; r_A)$, and Bob outputs $v = g(y, m_1, \ldots, m_i; r_A)$.

A *protocol with black-box access to a non-interactive functionality P* can be defined in a similar way. Here, the players have additionally the possibility to send messages to the functionality Q, which calculates a result according to the definition of Q, and sends it back to the players.

We will mostly be looking at the *semi-honest model*, where both players behave honestly, but may save all the

S. Wolf is with the Computer Science Department, ETH Zürich, CH-8092 Zürich, Switzerland. E-mail: wolf@inf.ethz.ch

J. Wullschleger is with the Department of Mathematics, University of Bristol, Bristol BS8 1TW, United Kingdom. E-mail: j.wullschleger@bristol.ac.uk

information they get during the protocol to obtain extra information about the other player's input or output. So, a dishonest Alice will output $(x, m_1, \ldots, m_i; r_A)$ instead of $f(x, m_1, \ldots, m_{i-1}; r_A)$.

A protocol (f, g) that implements a functionality P is *secure in the semi-honest model* for the Bob, if there exist a randomized function S for Alice, called the simulator, such that for every input (x, y) and for $(u, v) = h'(x, y)$, the distribution of $(S(u), v)$ is equal to the distribution of $((x, m_1, \ldots, m_i; r_A), g(y, m_1, \ldots, m_{i-1}; r_B))$.

A. Entropies and Information

The *conditional Shannon entropy* of X given Y is defined as[1]

$$\mathrm{H}(X \mid Y) = -\sum_{x,y} P_{XY}(x,y) \log P_{X|Y}(x \mid y) \ .$$

We will also use the notation

$$h(p) = -p \log p - (1-p) \log(1-p) \ ,$$

i.e., $h(p)$ is the Shannon entropy of a binary random variable that takes on one value with probability p and the other with $1-p$. The *conditional mutual information* is defined as

$$\begin{aligned} \mathrm{I}(X; Y \mid Z) &= \sum_{x,y,z} P_{XYZ}(x,y,z) \log \frac{P_{XY|Z}(x,y|z)}{P_{X|Z}(x|z) P_{Y|Z}(y|z)} \\ &= \mathrm{H}(X \mid Z) - \mathrm{H}(X \mid YZ) \ . \end{aligned}$$

We will need the following monotonicity inequalities:

$$\begin{aligned} \mathrm{H}(XY \mid Z) \geq \mathrm{H}(X \mid Z) \geq \mathrm{H}(X \mid YZ) \ , \\ \mathrm{I}(WX; Y \mid Z) \geq \mathrm{I}(X; Y \mid Z) \geq \mathrm{I}(X; Y \mid f(X)Z) \end{aligned}$$

for every function f.

The min- and max-entropies of X given Y are defined as

$$\begin{aligned} \mathrm{H}_{\min}(X \mid Y) &:= \min_{x,y}(-\log P_{X|Y}(x \mid y)) \ , \\ \mathrm{H}_{\max}(X \mid Y) &:= \max_{y} \log |\{x \in \mathcal{X} : P_{X|Y}(x \mid y) > 0\}| \ . \end{aligned}$$

The monotonicity of $\mathrm{H}_{\max}$,

$$\mathrm{H}_{\max}(XY \mid Z) \geq \mathrm{H}_{\max}(X \mid Z) \geq \mathrm{H}_{\max}(X \mid YZ)$$

follows from

$$\begin{aligned} &\max_{z} |\{(x,y) \in \mathcal{X} \times \mathcal{Y} : P_{XY|Z}(x,y \mid z) > 0\}| \\ &\quad \geq \max_{z} |\{x \in \mathcal{X} : P_{X|Z}(x \mid z) > 0\}| \\ &\quad \geq \max_{y,z} |\{x \in \mathcal{X} : P_{X|YZ}(x \mid y,z) > 0\}| \ . \end{aligned}$$

The monotonicity of $\mathrm{H}_{\min}$,

$$\mathrm{H}_{\min}(XY \mid Z) \geq \mathrm{H}_{\min}(X \mid Z) \geq \mathrm{H}_{\min}(X \mid YZ) \ .$$

follows from

$$\begin{aligned} &\max_{x,y,z} P_{XY|Z}(x, y \mid z) \\ &\quad \leq \max_{x,z} P_{X|Z}(x \mid z) \\ &\quad = \max_{x,z} \sum_{y} P_Y(y) P_{X|YZ}(x \mid y,z) \\ &\quad \leq \max_{x,z} \sum_{y} P_Y(y) \max_{y} P_{X|YZ}(x \mid y,z) \\ &\quad = \max_{x,y,z} P_{X|YZ}(x \mid y,z) \ . \end{aligned}$$

Furthermore, we will need the following property of all these measures: For (X_0, Y_0, Z_0) independent from (X_1, Y_1, Z_1), and $X = (X_0, X_1)$, $Y = (Y_0, Y_1)$, $Z = (Z_0, Z_1)$, we have

$$\begin{aligned} \mathrm{H}(X \mid Y) &= \mathrm{H}(X_0 \mid Y_0) + \mathrm{H}(X_1 \mid Y_1) \ , \\ \mathrm{H}_{\min}(X \mid Y) &= \mathrm{H}_{\min}(X_0 \mid Y_0) + \mathrm{H}_{\min}(X_1 \mid Y_1) \ , \\ \mathrm{H}_{\max}(X \mid Y) &= \mathrm{H}_{\max}(X_0 \mid Y_0) + \mathrm{H}_{\max}(X_1 \mid Y_1) \ , \\ \mathrm{I}(X; Y \mid Z) &= \mathrm{I}(X_0; Y_0 \mid Z_0) + \mathrm{I}(X_1; Y_1 \mid Z_1) \ . \end{aligned}$$

B. Common Part

Roughly speaking, the common part $X \wedge Y$ of X and Y is the maximal element of the set of all random variables (i.e., the *finest* random variable) that can be generated both from X and from Y without any error. For example, if $X = (X_0, X_1) \in \{0,1\}^2$ and $Y = (Y_0, Y_1) \in \{0,1\}^2$, and we have $X_0 = Y_0$ and $\Pr[X_1 \neq Y_1] = \varepsilon > 0$, then the common part of X and Y is equivalent to X_0. The common part was first introduced in [17]; in a cryptographic context, it was used in [25].

Definition 1: Let X and Y be random variables over $\mathcal{X}$ and $\mathcal{Y}$ and distributed according to P_{XY}. Then $X \wedge Y$, the *common part* of X and Y, is constructed in the following way:

- Consider the bipartite graph G with vertex set $\mathcal{X} \cup \mathcal{Y}$, and where two vertices $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ are connected by an edge if $P_{XY}(x,y) > 0$ holds.
- Let $f_X : \mathcal{X} \rightarrow 2^{\mathcal{X} \cup \mathcal{Y}}$ be the function that maps a vertex $v \in \mathcal{X}$ of G to the set of vertices in the connected component of G containing v. Let $f_Y : \mathcal{Y} \rightarrow 2^{\mathcal{X} \cup \mathcal{Y}}$ be the function that does the same for a vertex $w \in \mathcal{Y}$ of G.
- $X \wedge Y :\equiv f_X(X) \equiv f_Y(Y)$.

We will be needing the following two lemmas from [25].

Lemma 1: For all X, Y, and $\overline{C}$ for which there exist functions $\overline{f}_X$ and $\overline{f}_Y$ such that $\overline{C} = \overline{f}_X(X) = \overline{f}_Y(Y)$ holds, there exists a function g with $\overline{C} = g(X \wedge Y)$.

Proof: Let us assume that such a function g does not exist. Then there must exist values x_1 and x_2 with $\overline{f}_X(x_1) \neq \overline{f}_X(x_2)$ but $f_X(x_1) = f_X(x_2)$. Then x_1 and x_2 must be in the same connected component of the graph from Definition 1, and we can find values x_1', x_2', and y with $\overline{f}_X(x_1') \neq \overline{f}_X(x_2')$, $P_{XY}(x_1', y) > 0$, and $P_{XY}(x_2', y) > 0$. This implies that there cannot exist a function $\overline{f}_Y$ with $C \equiv \overline{f}_X(X) \equiv \overline{f}_Y(Y)$. ■

Lemma 2: Let (X_1, Y_1) and (X_2, Y_2) be independent. Then

$$(X_1 X_2) \wedge (Y_1 Y_2) \equiv (X_1 \wedge Y_1)(X_2 \wedge Y_2) \ .$$

[1] All logarithms throughout this article are binary.

Proof: We have $P_{X_1X_2Y_1Y_2}(x_1,x_2,y_1,y_2) > 0$ if and only if $P_{X_1Y_1}(x_1,y_1) > 0$ and $P_{X_2Y_2}(x_2,y_2) > 0$ because $P_{X_1X_2Y_1Y_2} = P_{X_1Y_1}P_{X_2Y_2}$. Hence, we have $f_{X_1X_2}(x_1,x_2) = f_{X_1X_2}(x'_1,x'_2)$ if and only if $f_{X_1}(x_1) = f_{X_1}(x'_1)$ and $f_{X_2}(x_2) = f_{X_2}(x'_2)$. ■

C. Sufficient Statistics

Intuitively speaking, the *sufficient statistics* of X from Y, denoted $X \searrow Y$, is the part of X that is correlated with Y. It has turned out to be a very useful concept in cryptography, where it was used in [24], [16], [18], [25].

Definition 2: Let X and Y be two random variables, and let $f(x) = P_{Y|X=x}$. The *sufficient statistics of X from Y* is defined as $X \searrow Y := f(X)$.

We will be needing the following lemma from [25].

Lemma 3: Let (X_1,Y_1) and (X_2,Y_2) be independent. Then

$$(X_1X_2) \searrow (Y_1Y_2) \equiv (X_1 \searrow Y_1)(X_2 \searrow Y_2) \ .$$

Proof: We have

$$P_{Y_1Y_2|(X_1,X_2)=(x_1,x_2)} = P_{Y_1|X_1=x_1}P_{Y_2|X_2=x_2} \ .$$

Hence,

$$P_{Y_1Y_2|(X_1,X_2)=(x_1,x_2)} \neq P_{Y_1Y_2|(X_1,X_2)=(x'_1,x'_2)}$$

if and only if either

$$P_{Y_1|X_1=x_1} \neq P_{Y_1|X_1=x'_1}$$

or

$$P_{Y_2|X_2=x_2} \neq P_{Y_2|X_2=x'_2}$$

holds. ■

III. Monotones

We will now present a general definition of a monotone for two-party computation and present several instances that satisfy our definition. A monotone is a function that takes a distribution of two random variables X and Y as input and outputs a number. In any protocol that Alice and Bob may execute, they cannot increase the value of any monotone. The monotone therefore tells us which results are impossible to achieve by a protocol, and its value should yield an upper bound on the ”usefulness” or potential of the random variables X and Y.

Definition 3: A *monotone for two-party computation* is a function $m(X,Y)$, such that

1. for all X, Y, and Z with $X \longleftrightarrow Y \longleftrightarrow Z$, we have
$$m((X,Y),Z) \leq m(Y,Z)$$
and
$$m(X,(Y,Z)) \leq m(X,Y) \ ,$$
2. for all X, Y, and all functions f, we have
$$m((X,f(Y)),Y) \leq m(X,Y)$$
and
$$m(X,(Y,f(X))) \leq m(X,Y) \ .$$
3. for all W, X, Y, and Z, with $W \longleftrightarrow X \longleftrightarrow Y$ and $X \longleftrightarrow Y \longleftrightarrow Z$, we have
$$m((W,X),(Y,Z)) \geq m(X,Y) \ .$$
4. For independent pairs (X_0,Y_0) and (X_1,Y_1) we have
$$m((X_0,X_1),(Y_0,Y_1)) = m(X_0,Y_0) + m(X_1,Y_1) \ .$$

In order to define functions that are monotones we will make use of the *common part* and the *sufficient statistics*. The following lemma shows that under local data processing, i.e., creation of additional randomness, these values are invariant. It will help us to show that our functions satisfy Condition 1 of Definition 3.

Lemma 4: Let X, Y, and Z be random variables with $X \longleftrightarrow Y \longleftrightarrow Z$. Then we have

$$\begin{aligned} X \searrow (Y,Z) &\equiv X \searrow Y \ , \\ (Y,Z) \searrow X &\equiv Y \searrow X \ , \\ X \wedge (Y,Z) &\equiv X \wedge Y \ . \end{aligned}$$

Proof:

- We have $P_{YZ|X=x} = P_{Y|X=x}P_{Z|Y}$. Therefore, for all $x,x' \in \mathcal{X}$, the function $P_{YZ|X=x}$ is different from $P_{YZ|X=x'}$ if and only if $P_{Y|X=x}$ is different from $P_{Y|X=x'}$.
- We have $P_{X|Y=y,Z=z} = P_{X|Y=y}$. Therefore, for all $z,z' \in \mathcal{Z}$ and $y,y' \in \mathcal{Y}$, the function $P_{X|Y=y,Z=z}$ is different from $P_{X|Y=y',Z=z'}$ if and only if $P_{Y|X=x}$ is different from $P_{Y|X=x'}$.
- We have $P_{XYZ} = P_{XY}P_{Z|Y}$. Let us look at the connection graph between all the values x and (y,z) for which $P_X(x) > 0$ and $P_{YZ}(y,z) > 0$ hold. Then x and (y,z) are connected if and only if $P_{XYZ}(x,y,z) > 0$ holds. Since $P_{Z|Y}(z \mid y) > 0$, this holds if and only if $P_{XY}(x,y) > 0$ holds. Hence, $X \wedge (Y,Z) \equiv X \wedge Y$.

■

To show that our functions satisfy Condition 2 of Definition 3, we need the following lemma.

Lemma 5: Let X and Y be two random variables and f a function. There exist functions g and g' such that

$$\begin{aligned} X \searrow (Y,f(X)) &\equiv (X \searrow Y, f(X)) \ , \\ (Y,f(X)) \searrow X &= g((Y \searrow X, f(X))) \ , \\ g'(X \wedge (Y,f(X))) &= (X \wedge Y, f(X)) \ . \end{aligned}$$

Proof:

- Let $h_1(X) := X \searrow (Y,f(X))$ and $h_2(X) := (X \searrow Y, f(X))$, and let $F = f(X)$. We have $P_{YF|X} = P_{Y|X}P_{F|X}$. For all x,x' with $h_1(x) = h_1(x')$, we have $P_{YF|X=x} = P_{YF|X=x'}$, which holds exactly if $P_{Y|X=x} = P_{Y|X=x'}$ and $f(x) = f(x')$ hold, which is equivalent to $h_2(x) = h_2(x')$. Hence, $X \searrow (Y,f(X)) \equiv (X \searrow Y, f(X))$.
- Let $h_1(X,Y) := (Y,f(X)) \searrow X$ and $h_2(X,Y) := (Y \searrow X, f(X))$. For all x,x',y, and y' with $h_2(x,y) = h_2(x',y')$, we have $P_{X|Y=y} = P_{X|Y=y'}$ and $f(x) = f(x')$. It follows $P_{X|Y=y,f(X)=f(x)} = P_{X|Y=y',f(X)=f(x')}$, and, hence, $h_1(x,y) = h_1(x',y')$. Therefore, there must exist a function g with $h_1 = g \circ h_2$.

- The statement follows from Lemma 1 and from the fact that both $X \wedge Y$ and $f(X)$ can be calculated from both X and $(Y, f(X))$.

■

Lemma 6: For all random variables W, X, Y, and Z where $W \longleftrightarrow X \longleftrightarrow Y$ and $X \longleftrightarrow Y \longleftrightarrow Z$, we have

$$\mathrm{I}(WX;YZ \mid (WX) \wedge (YZ)) \geq \mathrm{I}(X;Y \mid X \wedge Y) .$$

Proof: We have $P_{WXYZ} = P_{XY} P_{WZ|XY}$. The channel $P_{WZ|XY}$ can be simulated by a function f that gets as input (X, Y) and the random value C that is independent of (X, Y), and outputs (W, Z). $(WX) \wedge (YZ)$ can be calculated from (X, C), as well as from (Y, C). Hence, it can also be calculated from $(X \wedge Y, C)$. Furthermore, $X \wedge Y$ can be calculated from $(WX) \wedge (YZ)$. Therefore, there exists a function g, such that

$$(WX) \wedge (YZ) = (X \wedge Y, g(X \wedge Y, C)) .$$

Since C is independent of X and Y, it follows that

$$\begin{aligned} \mathrm{I}(WX;YZ \mid (WX) \wedge (YZ)) &\geq \mathrm{I}(X;Y \mid (WX) \wedge (YZ)) \\ &= \mathrm{I}(X;Y \mid X \wedge Y) \end{aligned}$$

■

Lemma 7: Let W, X, Y, and Z be random variables, and let $W \longleftrightarrow X \longleftrightarrow Y$. There exist a function g such that

$$g((W,X) \searrow (Y,Z)) = X \searrow Y .$$

Proof: Let $h_1(W,X) := X \searrow Y$ and $h_2(W,X) := (W,X) \searrow (Y,Z)$. For all w, w', x, and x' with $h_2(w,x) = h_2(w',x')$, we have $P_{YZ|W=w,X=x} = P_{YZ|W=w',X=x'}$. Since $W \longleftrightarrow X \longleftrightarrow Y$, we have

$$P_{YZ|W=w,X=x} = P_{Y|X=x} P_{Z|Y,W=w,X=x} .$$

It follows $P_{Y|X=x} = P_{Y|X=x'}$, and, hence, $h_1(w,x) = h_1(w',x')$. Therefore, there must exist a function g with $h_1 = g \circ h_2$. ■

Lemma 8: Let W, X, Y, and Z be random variables, and let $W \longleftrightarrow X \longleftrightarrow Y$ and $X \longleftrightarrow Y \longleftrightarrow Z$. For all $\mathrm{H}^* \in \{\mathrm{H}, \mathrm{H}_{\min}, \mathrm{H}_{\max}\}$, we have

$$\mathrm{H}^*((WX) \searrow (YZ) \mid YZ) \geq \mathrm{H}(X \searrow Y \mid Y) .$$

Proof: From Lemmas 7 follows that

$$\mathrm{H}^*((WX) \searrow (YZ) \mid YZ) \geq \mathrm{H}^*(X \searrow Y \mid YZ) ,$$

and from $X \longleftrightarrow Y \longleftrightarrow Z$ follows that

$$\mathrm{H}^*(X \searrow Y \mid YZ) = \mathrm{H}^*(X \searrow Y \mid Y) .$$

■

We are now ready to present our monotones for two-party computation.

Theorem 1: For all $\mathrm{H}^* \in \{\mathrm{H}, \mathrm{H}_{\min}, \mathrm{H}_{\max}\}$, the functions $\mathrm{H}^*(X \searrow Y \mid Y)$, $\mathrm{H}^*(Y \searrow X \mid X)$, and $\mathrm{I}(X;Y \mid X \wedge Y)$ are monotones for two-party computation.

Proof: For $X \longleftrightarrow Y \longleftrightarrow Z$ and for all W, we have

$$\mathrm{I}(XY;Z \mid W) = \mathrm{I}(Y;Z \mid W) .$$

From Lemma 4 and monotonicity follows that all these functions satisfy Condition 1. Using Lemma 5 and monotonicity, we obtain

$$\begin{aligned} &\mathrm{H}^*((Y, f(X)) \searrow X \mid X) \\ &\quad \leq \mathrm{H}^*((Y \searrow X, f(X)) \mid X) \\ &\quad = \mathrm{H}^*(Y \searrow X \mid X) , \end{aligned}$$

$$\begin{aligned} &\mathrm{H}^*(X \searrow (f(X), Y) \mid f(X), Y) \\ &\quad = \mathrm{H}^*((X \searrow Y, f(X)) \mid f(X), Y) \\ &\quad = \mathrm{H}^*(X \searrow Y \mid f(X), Y) \\ &\quad \leq \mathrm{H}^*(X \searrow Y \mid Y) , \end{aligned}$$

and

$$\begin{aligned} &\mathrm{I}(X; (f(X), Y) \mid X \wedge (f(X), Y)) \\ &\quad \leq \mathrm{I}(X; (f(X), Y) \mid f(X), X \wedge Y) \\ &\quad = \mathrm{I}(X; Y \mid f(X), X \wedge Y) \\ &\quad \leq \mathrm{I}(X; Y \mid X \wedge Y) , \end{aligned}$$

from which Condition 2 follows. Condition 3 follows from Lemmas 6 and 8. Condition 4 follows from Lemmas 2 and 3 and the properties of H and I. ■

Depending on the situation, some of these monotones give better bounds than others. If many independent instances of the underlying resource are given, then the monotones using H should be used. When the protocol makes some *extraction*, i.e., transforms a resource into (almost) uniform randomness (a very common goal of protocols), then the two monotones using $\mathrm{H}_{\min}$ might be preferable to H. The monotone using $\mathrm{H}_{\max}$ gives better bounds for *simulation* protocols, i.e., protocols that use random bits as a resource and have a non-uniform output. The monotone using I is much less intuitive than the others, but nevertheless yields, in some situations, a better bound than all the others. For an example of this, see Corollary 1.

IV. Lower Bounds on Reductions of Two-Party Primitives

In this section we show how all these monotones can be used to get lower bounds on the efficiency of reductions among a variety of two-party primitives. These primitives must satisfy the following property: In the honest-but-curious model, they must be *equivalent* to a primitive *without any inputs*, i.e., distributed randomness. By "equivalent," we mean that there exist two protocols: One that generates the distributed randomness using one instance of the primitive, and the other implementing the primitive, using the distributed randomness. Both protocols must be secure in the semi-honest model. For any such primitive, and any monotone m, we will now define the *m-capacity* for two-party computation.

Definition 4: Let m be a monotone for two-party computation. Let P be a primitive between two players Alice and Bob. Furthermore, let P be equivalent to a primitive P' which does not have any inputs, and outputs random variables X and Y. Then the *m-capacity of P* is defined by

$$C_m(P) := m(X, Y) .$$

If there does not exist such a P', then the m-capacity is undefined.

Theorem 2: Let m be a monotone for two-party computation. Let $P_1, \ldots, P_n$, and P be primitives for which the m-capacity is defined. If there exists a protocol π that securely implements P using $P_1, \ldots, P_n$ in the semi-honest model (where every primitives P_i is independent of the others and can only be used once), then

$$C_m(P) \leq \sum_i C_m(P_i) \ .$$

Proof: Let (X_i, Y_i) be the distributed randomness which is equivalent to P_i, and let $(\overline{X}, \overline{Y})$ be the output of P', the primitive which is equivalent to P. Let $X^n = (X_1, \ldots, X_n)$ and $Y^n = (Y_1, \ldots, Y_n)$. Since (X_i, Y_i) is independent from (X_j, Y_j), we have

$$m(X^n, Y^n) = \sum_i m(X_i, Y_i) \ .$$

Let π' be defined as follows: It takes X^n and Y^n as input, transform them into the primitives $P_1, \ldots P_n$, applies π, and transforms P into P'. Let X and Y be the outputs and U and V be the entire views of A and B, respectivly, after the execution of π'. Let U_i and V_i be memory of Alice and Bob after step i of π'. After the first step, we have $U_1 = X^n$ and $V_1 = Y^n$. Since m is a monotone for two-party computation, we have for all i that $m(U_{i+1}, V_{i+1}) \leq m(U_i, V_i)$ and therefore

$$m(U, V) \leq m(X^n, Y^n) \ .$$

Since π is secure, also π' is secure. Therefore, (X, Y) has the same distribution as $(\overline{X}, \overline{Y})$, and there exist simulators S_A and S_B such that (U, Y) has the same distribution as $(S_A(\overline{X}), \overline{Y})$, and (X, V) has the same distribution as $(\overline{X}, S_B(\overline{Y}))$. It follows that we have $U \longleftrightarrow X \longleftrightarrow Y$ and $X \longleftrightarrow Y \longleftrightarrow V$. From the definition of m follows that $m(U, V) \geq m(\overline{X}, \overline{Y})$. Hence,

$$\begin{aligned} C_m(P) = m(\overline{X}, \overline{Y}) &\leq m(U, V) \leq m(X^n, Y^n) \\ = \sum_i m(X_i, Y_i) &= \sum_i C_m(P_i) \ . \end{aligned}$$

■

V. Lower Bounds for Oblivious-Transfer Reductions

We now apply the results of the last sections for deriving lower bounds on oblivious-transfer reductions. We use the following three monotones:

$$\begin{aligned} A(X, Y) &:= \mathrm{H}(X \searrow Y \mid Y) \ , \\ B(X, Y) &:= \mathrm{H}(Y \searrow X \mid X) \ , \\ C(X, Y) &:= \mathrm{I}(X; Y \mid X \wedge Y) \ . \end{aligned}$$

In m-out-of-n k-string oblivious transfer, denoted $\binom{n}{m}$-OT^k, the sender inputs n k-bit messages out of which the receiver can choose to read m but does not obtain any further information about the messages; the sender, on the other hand, does not obtain any information on the receiver's choice. In [1], it has been shown that $\binom{2}{1}$-OT^1 is *equivalent* to pieces of information with a certain distribution (in other words, oblivious transfer can be pre-computed and stored). This result generalizes to $\binom{n}{m}$-OT^k in a straight-forward way.

$t \geq \ldots$	$K \geq k$	$K < k$
$N \geq n$	$\frac{(N-1)K}{(n-1)k}$	$\max\left(\frac{(N-1)K}{(n-1)k}, \frac{\log N}{\log n}\right)$
$N < n$	$\frac{K}{k}$	1

Fig. 1. Lower bounds on the number t of instances of $\binom{n}{1}$-OT^k needed to implement $\binom{N}{1}$-OT^K.

Lemma 9: Let $P = \binom{n}{m}$-OT^k. Then, we have

$$C_A(P) = (n-m)k, \ \ C_B(P) = \log \binom{n}{m}, \ \ C_C(P) = mk \ .$$

We can now easily obtain lower bounds on the reducibility between different variants of oblivious transfer. The bound of Corollary 1 is an improvement on an earlier bound by Dodis and Micali [15].

Corollary 1: Assume that there exists a protocol for realizing unconditionally secure $\binom{N}{M}$-OT^K from t instances of $\binom{n}{m}$-OT^k. Then we have

$$t \ \geq \ \max\left(\frac{(N-M)K}{(n-m)k}, \frac{\log\binom{N}{M}}{\log\binom{n}{m}}, \frac{MK}{mk}\right) \ .$$

Proof: Follows from Lemma 9 and Theorem 2. ■

For the special case $M = m = 1$, the obtained bounds are shown in Figure 1.

We can now also easily prove that OT cannot be extended, i.e., given s instances of OT, there does not exist a protocol that constructs $s+1$ instances of OT. This was first proved by Beaver in [2]. This implies that it is impossible to implement OT from scratch.

Corollary 2: There cannot exist a protocol that implements $s+1$ instances of $\binom{n}{m}$-OT^k out of s instances of $\binom{n}{m}$-OT^k.

Let us now look at reductions of OT to noisy channels.

Definition 5: Let $0 < \varepsilon < \frac{1}{2}$. The *binary symmetric noisy channel with error* ε, or (ε)-BNC, is defined as follows. First, it waits for A to send an input $x \in \{0, 1\}$. After receiving x, it outputs a value $Y \in \{0, 1\}$ to B, where $\Pr[Y \neq x] = \varepsilon$.

We can easily show that a binary noisy channel is equivalent to distributed randomness: Alice sends a random bit, which is received by Bob with an error ε. Later, Alice can send her bit xored with this random bit, and Bob will get to know her bit with probability ε.

Lemma 10:

$$\begin{aligned} C_A((\varepsilon)\text{-}\mathsf{BNC}) &= h(\varepsilon) \ , \\ C_B((\varepsilon)\text{-}\mathsf{BNC}) &= h(\varepsilon) \ , \\ C_C((\varepsilon)\text{-}\mathsf{BNC}) &= 1 - h(\varepsilon) \ . \end{aligned}$$

Corollary 3: If a protocol implements $\binom{N}{M}$-OT^K from t instances of (ε)-BNC in the semi-honest model, then

$$t \geq \max\left(\frac{(N-M)K}{h(\varepsilon)}, \frac{\log\binom{N}{M}}{h(\varepsilon)}, \frac{MK}{1-h(\varepsilon)}\right) \ .$$

Definition 6: Rabin oblivious transfer, or (p)-$\mathsf{RabinOT}$, is defined as follows. First, it waits for Alice to send an input $x \in \{0,1\}$. After receiving x, it outputs a value $Y \in \{0,1,\perp\}$ to Bob, where $\Pr[Y = x] = p$ and $\Pr[Y = \perp] = 1 - p$.

In the semi-honest model, (p)-$\mathsf{RabinOT}$ is equivalent to a primitive without any input. Alice sends a random bit using (p)-$\mathsf{RabinOT}$ that is received by Bob with probability p. Later, Alice can send her bit xored with this random bit.

Lemma 11:

$$\begin{aligned} C_A((p)\text{-}\mathsf{RabinOT}) &= 1 - p \,, \\ C_B((p)\text{-}\mathsf{RabinOT}) &= h(p) \,, \\ C_C((p)\text{-}\mathsf{RabinOT}) &= p \,. \end{aligned}$$

Corollary 4: If a protocol implements $\binom{N}{M}$-OT^K from t instances of (p)-$\mathsf{RabinOT}$ in the semi-honest model, then

$$t \geq \max\left(\frac{(N-M)K}{1-p}, \frac{\log\binom{N}{M}}{h(p)}, \frac{MK}{p}\right) .$$

Note that for some special cases, we know reductions that achieve the optimal bound of Corollaries 1 and 4, even in the malicious model [15], [26], [11]. For Corollary 3, however, an efficient reduction is still missing.

The bounds of this section can easily be generalized in many ways. For example, one could also consider the reduction of OT to different variants of OT, binary noisy channels, and Rabin-OTs *at the same time*.

VI. Conclusions

We have presented several monotones for two-party computation and have shown that they provide us with a powerful tool to derive lower bounds for reductions between functionalities in the semi-honest model. Note that such lower bounds do generally *not* directly imply lower bounds in the malicious model, as there are functionalities which can trivially be implemented in the malicious model, but not in the semi-honest model. However, for OT the lower bounds do also apply to the malicious model, as it can be shown that any secure implementation of OT in the malicious model is also secure in the semi-honest model.

In our work we have only considered *perfect* reductions, however it would be preferable to have lower bounds for reductions with a (small) probability of error.

Acknowledgments

The authors thank Don Beaver, Claude Crépeau, Thomas Holenstein, Anderson Nascimento, and Renato Renner for interesting discussions on the subject of this paper, and three anonymous reviewers for their helpful comments on an earlier version. This work was supported by Switzerland's SNF, Canada's NSERC, and Québec's FQRNT.

References

[1] D. Beaver. Precomputing oblivious transfer. In *Advances in Cryptology — EUROCRYPT '95*, vol. 963 of *LNCS*, pp. 97–109. Springer-Verlag, 1995.

[2] D. Beaver. Correlated pseudorandomness and the complexity of private computations. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC '96)*, pp. 479–488. ACM Press, 1996.

[3] G. Brassard, C. Crépeau, and S. Wolf. Oblivious transfers and privacy amplification. *Journal of Cryptology*, 16(4):219–237, 2003.

[4] C. Cachin. On the foundations of oblivious transfer. In *Advances in Cryptology — EUROCRYPT '98*, vol. 1403 of *LNCS*, pp. 361–374. Springer-Verlag, 1998.

[5] N. J. Cerf, S. Massar, and S. Schneider. Multipartite classical and quantum secrecy monotones. *Phys. Rev. A*, 66:042309.1–042309.13, 2002.

[6] C. Crépeau. *Correct and private reductions among oblivious transfers*. PhD thesis, Massachusetts Institute of Technology, USA, 1990.

[7] C. Crépeau. Efficient cryptographic protocols based on noisy channels. In *Advances in Cryptology — CRYPTO '97*, vol. 1233 of *LNCS*, pp. 306–317. Springer-Verlag, 1997.

[8] C. Crépeau and J. Kilian. Achieving oblivious transfer using weakened security assumptions (extended abstract). In *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science (FOCS '88)*, pp. 42–52, 1988.

[9] C. Crépeau, K. Morozov, and S. Wolf. Efficient unconditional oblivious transfer from almost any noisy channel. In *Proceedings of Fourth Conference on Security in Communication Networks (SCN)*, vol. 3352 of *LNCS*, pp. 47–59. Springer-Verlag, 2004.

[10] C. Crépeau and M. Sántha. On the reversibility of oblivious transfer. In *Advances in Cryptology — EUROCRYPT '91*, vol. 547 of *LNCS*, pp. 106–113. Springer, 1991.

[11] C. Crépeau and G. Savvides. Optimal reductions between oblivious transfers using interactive hashing. In *Advances in Cryptology — EUROCRYPT '06*, vol. 4004 of *LNCS*, pp. 201–221. Springer-Verlag, 2006.

[12] I. Damgård, S. Fehr, K. Morozov, and L. Salvail. Unfair noisy channels and oblivious transfer. In *Theory of Cryptography Conference — TCC '04*, vol. 2951 of *LNCS*, pp. 355–373. Springer-Verlag, 2004.

[13] I. Damgård, S. Fehr, L. Salvail, and C. Schaffner. Oblivious transfer and linear functions. In *Advances in Cryptology — CRYPTO '06*, vol. 4117 of *LNCS*. Springer-Verlag, 2006.

[14] I. Damgård, J. Kilian, and L. Salvail. On the (im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions. In *Advances in Cryptology — EUROCRYPT '99*, vol. 1592 of *LNCS*, pp. 56–73. Springer-Verlag, 1999.

[15] Y. Dodis and S. Micali. Lower bounds for oblivious transfer reductions. In *Advances in Cryptology — EUROCRYPT '99*, vol. 1592 of *LNCS*, pp. 42–55. Springer-Verlag, 1999.

[16] M. Fitzi, S. Wolf, and J. Wullschleger. Pseudo-signatures, broadcast, and multi-party computation from correlated randomness. In *Advances in Cryptology — CRYPTO '04*, vol. 3152 of *LNCS*, pp. 562–578. Springer-Verlag, 2004.

[17] P. Gacs and J. Körner. Common information is far less than mutual information. *Probl. Contr. Inform. Theory*, 2:149–162, 1973.

[18] H. Imai, J. Müller-Quade, A. Nascimento, and A. Winter. Rates for bit commitment and coin tossing from noisy correlation. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT '04)*, 2004.

[19] U. Maurer. Information-theoretic cryptography. In *Advances in Cryptology — CRYPTO '99*, vol. 1666 of *LNCS*, pp. 47–64. Springer-Verlag, 1999.

[20] U. Maurer and S. Wolf. Unconditionally secure key agreement and the intrinsic conditional information. *IEEE Transaction on Information Theory*, 45(2):499–514, 1999.

[21] A. Nascimento and A. Winter. On the oblivious transfer capacity of noisy correlations. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT '06)*, 2006.

[22] R. Ostrovsky, R. Venkatesan, and M. Yung. Fair games against an all-powerful adversary. In *Advances in Computational Complexity Theory*, vol. 13 of *AMS DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 155–169. AMS, 1993.

[23] R. Renner and S. Wolf. New bounds in secret-key agreement: The gap between formation and secrecy extraction. In *Advances in Cryptology — EUROCRYPT '03*, vol. 2656 of *LNCS*, pp. 562–577. Springer-Verlag, 2003.

[24] A. Winter, A. C. A. Nascimento, and H. Imai. Commitment capacity of discrete memoryless channels. In *IMA Int. Conf.*, pp. 35–51, 2003.

[25] S. Wolf and J. Wullschleger. Zero-error information and applications in cryptography. In *Proceedings of 2004 IEEE Information Theory Workshop (ITW '04)*, 2004.

[26] S. Wolf and J. Wullschleger. New monotones and lower bounds in unconditional two-party computation. In *Advances in Cryptology — CRYPTO '05*, vol. 3621 of *LNCS*, pp. 467–477, 2005.

[27] S. Wolf and J. Wullschleger. Oblivious transfer is symmetric. In *Advances in Cryptology — EUROCRYPT '06*, vol. 4004 of *LNCS*, pp. 222–232. Springer-Verlag, 2006.

[28] J. Wullschleger. Oblivious-transfer amplification. In *Advances in Cryptology — EUROCRYPT '07*, vol. 4515 of *LNCS*, pp. 555–572. Springer-Verlag, 2007.