Completeness in Two-Party Secure Computation - A Computational View

Danny Harnik[*] Moni Naor[†] Omer Reingold[‡] Alon Rosen[§]

Abstract

A Secure Function Evaluation (SFE) of a two-variable function $f(\cdot,\cdot)$ is a protocol that allows two parties with inputs x and y to evaluate $f(x,y)$ in a manner where neither party learns "more than is necessary". A rich body of work deals with the study of completeness for secure two-party computation. A function f is *complete* for SFE if a protocol for securely evaluating f allows the secure evaluation of all (efficiently computable) functions. The questions investigated are which functions are complete for SFE, which functions have SFE protocols unconditionally and whether there are functions that are neither complete nor have efficient SFE protocols.

The previous study of these questions was mainly conducted from an Information Theoretic point of view and provided strong answers in the form of combinatorial properties. However, we show that there are major differences between the information theoretic and computational settings. In particular, we show functions that are considered as having SFE unconditionally by the combinatorial criteria but are actually complete in the computational setting.

We initiate the fully computational study of these fundamental questions. Somewhat surprisingly, we manage to provide an almost full characterization of the complete functions in this model as well. More precisely, we present a *computational criterion* (called *computational row non-transitivity*) for a function f to be complete for the asymmetric case. Furthermore, we show a matching criterion called *computational row transitivity* for f to have a simple SFE (based on no additional assumptions). This criterion is close to the negation of the computational row non-transitivity and thus we essentially characterize all "nice" functions as either complete or having SFE unconditionally.

[*] Department of Computer Science and Applied Mathematics, Weizmann Institute, Rehovot, 76100 Israel. mail: harnik@wisdom.weizmann.ac.il. Research supported in part by a grant from the Israel Science Foundation.

[†] Incumbent of the Judith Kleeman Professorial Chair, Department of Computer Science and Applied Math, Weizmann Institute of Science, Rehovot 76100 Israel. Email: naor@wisdom.weizmann.ac.il. Research supported in part by a grant from the Israel Science Foundation.

[‡] AT&T Labs - Research. Room A201, 180 Park Avenue, Bldg. 103, Florham Park, NJ, 07932, USA. E-mail: omer@research.att.com. Part of this research was performed while visiting the Institute for Advanced Study, Princeton, NJ.

[§] Laboratory for Computer Science, Massachusetts Institute of Technology. Email: alon@lcs.mit.edu. Part of this work done while at the Weizmann Institute of Science, Israel.

1 Introduction

A Secure Function Evaluation (SFE) of a two-variable function $f(\cdot,\cdot)$ is a protocol between two parties Alice and Bob, where Alice holds an input x and Bob holds an input y. Loosely speaking, at the end of the protocol Alice should learn the value $f(x, y)$ but learn nothing more (other than what can be efficiently deduced from x and $f(x, y)$). Bob, on the other hand, must learn nothing.

There are various definitions and models for SFE, and indeed the above definition describes just one of them. For example, one may consider a symmetric version where both parties learn the value $f(x, y)$. However, in this work we choose to concentrate on this asymmetric version, where only Alice receives the output.[1] Furthermore, at this point we conduct our work in the "semi-honest" model where Alice and Bob follow the protocol honestly, but may later try to extract more information from the transcript of the protocol.[2] We refer the reader to [19] for an extensive overview of the various possible models and definitions of SFE.

1.1 Completeness in Secure Function Evaluation

A major component in the construction of SFE protocols is the Oblivious Transfer protocol (OT), Rabin's brainchild [42]. OT refers to several different versions of SFE protocols, all of which turned out to be equivalent. For instance, consider the 1-2-OT [14], where Bob has two secret bits b_0, b_1 and Alice has a choice bit c. At the end of the protocol Alice learns b_c but learns nothing about b_{1-c}, while Bob learns nothing about Alice's choice. This can be viewed as an SFE protocol for the function $f_{\mathrm{OT}}(c,(b_0, b_1)) = b_c$.

OT plays a key role in secure computation since it was shown to be *complete for SFE* [45, 22, 30], i.e. the SFE of every efficiently computable function f can be efficiently reduced to OT. In other words, an SFE protocol for f can be constructed using calls to an OT protocol.[3] Indeed, several implementations of OT protocols have been suggested relying on various computational assumptions (see Section 2.3 for more details).

The fact that there exists a simple complete function for secure evaluation is intriguing in its own right and has led to the natural question of what other functions are complete. Let us denote by $\mathcal{SFE}$-C the set of functions which are complete for SFE. This set in particular contains the function f_{OT}. Let us also denote by Eff-$\mathcal{SFE}$ the set of functions for which there exists (efficient) SFE. The set Eff-$\mathcal{SFE}$ is also non empty as there are functions for which trivial SFE exists (such as functions $f(x, y)$ which only depend on x). There are many fundamental open problems regarding these sets, in particular, the following questions are the foci of our paper.

1. Which functions other than f_{OT} are complete for SFE? Is there a natural way of characterizing the functions in $\mathcal{SFE}$-C? The ability of identifying functions in $\mathcal{SFE}$-C can be particularly useful as a tool for implementing OT: If we are able to design SFE for a function f which is $\mathcal{SFE}$-C we are immediately guaranteed an implementation of OT (we show a specific example in Section 1.6).

2. How do the two sets relate? One possibility which is consistent with our current knowledge is that $\mathcal{SFE}$-C = Eff-$\mathcal{SFE}$ and they both contain every efficiently computable function. This is indeed implied by the existence of OT. If however OT does not exist, we have that

[1] The choice of the asymmetric model can be justified by the fact that a symmetric protocol seems problematic to achieve (as brought forth in [2]). This is since the first party to receive its output may maliciously end the protocol, thus preventing the other party from learning its output.

[2] Secure protocols in the semi-honest model can later be transformed to be secure in a malicious model as well (see Section 1.5).

[3] The notions of completeness and reductions are made formal in Section 2.2.

	y_0	y_1
x_0	a	a
x_1	a	b

	y_0	y_1
x_0	a	a
x_1	b	c

Table 1: Imbedded Or Insecure Minor

$\mathcal{SFE}$-C$\cap$Eff-$\mathcal{SFE} = \emptyset$. In this case, the picture is not as clear: Are there interesting functions that can still have "non-trivial" SFE? Consider in this case the variety of possible assumptions of the sort $f \in$ Eff-$\mathcal{SFE}$ for functions $f \notin \mathcal{SFE}$-C. Are any of these assumptions useful "fall backs" in the unfortunate scenario where OT does not exist?

1.2 Related Work

Indeed the above questions were also discussed in a large body of work [4, 10, 34, 31, 2, 35, 33, 32, 15]. This study was mainly conducted from an Information Theoretic point of view. Specifically, in most of these papers SFE protects against computationally unbounded parties. Matching the definition of SFE, the notion of completeness is usually information theoretic as well. Most importantly, these papers usually only consider SFE of *"finite" functions.* That is, functions defined over a finite domain. As we will demonstrate below, when finite domain is interpreted as constant size domain, many of the computational concerns are avoided. From information theoretic perspective this line of research obtains very strong results. Loosely, these papers classify functions as either complete or in Eff-$\mathcal{SFE}$ unconditionally[4] and provide combinatorial properties for separating the two. Formalizing the various models considered by this line of work (and specifically the various notions of completeness and triviality) is beyond the scope of this paper. Instead, we concentrate here on the combinatorial criteria provided by these works and analyze their applicability to the computational setting:

Imbedded Or: The papers initiating this line of research are those of Chor and Kushilevitz [10, 34] and Kilian [31]. They consider the symmetric version of SFE and prove that a function f is complete iff it contains an *imbedded OR*. An imbedded OR of a function f consists of inputs x_0, x_1 and y_0, y_1 such that $f(x_0, y_0) = f(x_0, y_1) = f(x_1, y_0) \neq f(x_1, y_1)$ (see Table 1.2).[5] When discussing Boolean functions, it turns out that the non-complete functions (functions not containing an Imbedded Or) have trivial SFE protocols.

Insecure Minor: Beimel, Malkin and Micali [2] discuss the asymmetric model (which is the focus of our work). They defined an *insecure minor* to consist of inputs x_0, x_1 and y_0, y_1 such that $f(x_0, y_0) = f(x_0, y_1)$ but $f(x_1, y_0) \neq f(x_1, y_1)$ (shown in Table 1.2),[6] and showed that every finite function with an insecure minor is complete, while every other finite function has a trivial SFE. This provides a very complete answer to the questions we posed above with respect to finite functions. This work was conducted under computational definitions of security, using the compiler of GMW [22] to assure security against malicious parties rather than semi-honest parties (as we do in this work, see Section 1.5).

Other Work: Other works include the generalization of the criteria to the case of multi-parties (in the symmetric case) [35, 33], and probabilistic functionalities [32] (as opposed to deterministic

[4] By unconditionally we mean that no hardness assumption is needed.

[5] The name "imbedded OR", comes from the fact that the function projected to these four inputs looks like an OR or an AND gate.

[6] Note that every imbedded OR is also an insecure minor but not vice versa

functions). Completeness in multi party computation was also studied with regard to a measure of cardinality (the number of participating parties) [15].

1.3 Computational Considerations Make a Difference:

While the information theoretic approach gives very elegant and tight answers when no computational aspects are discussed (as the case is when discussing functions with a constant domain size), we stress that this is not satisfactory when computational considerations are taken into account. Particularly, functions with no insecure minor do not necessarily have efficient SFE protocols. In fact, such functions can even be complete!

The problem can be illustrated nicely by looking at a special case of functions with no insecure minor - the one-to-one functions.[7] By the BMM-criterion a one-one function f has an SFE unconditionally. This is justified by the following simple protocol: Let Bob send y to Alice. Indeed Bob learns nothing and Alice learns $f(x, y)$. Furthermore, Alice's view could be simulated from x and $f(x, y)$ as y is fully determined by these values. A-priory however, the running time of this simulator is exponential in n. This is acceptable if we think of n as small (as [2] did) but is impermissible when considering a large domain size. For general functions with no insecure minor (not necessarily one-to-one) the situation may even be worse. Not only may the simulator's running time be large, but the running time of the SFE protocol given by [2] may in itself be exponential in the input's length.

We show an example, where the insecure minor classification *does not hold* when considering a function on large input size under the assumption that one-to-one one-way functions exist.[8]

Example 1 *Let g be a one-to-one one-way function ($g : \{0,1\}^n \to \{0,1\}^n$ for all n). The function $f : \{0,1\}^1 \times \{0,1\}^{2n} \to \{0,1\}^{2n+1}$ is defined by:*

$$f(c, (y_0, y_1)) = (c, y_c, g(y_{1-c}))$$

As f is one-to-one it does not have an insecure minor. On the other hand, we claim that f is actually complete. As a proof sketch we present the following construction of OT from an SFE for f. Let Alice hold the choice bit c and Bob hold the secrets b_0, b_1. Alice and Bob run the SFE protocol for f with the bit c as Alice's input and random strings $y_0, y_1 \in \{0,1\}^n$ as Bob's inputs. Now Bob sends to Alice $b_0 \oplus h_g(y_0)$ and $b_1 \oplus h_g(y_1)$, where h_g is a hardcore bit of the function g (recall that g is one-way). Clearly Bob learns nothing from the protocol, since other than participating in the SFE he only sends information and receives none. Alice, on the other hand, learns y_c and hence she also learns $h_g(y_c)$ and subsequently the secret b_c (as required). However, due to the computational hardness of inverting g, Alice can't guess the bit b_{1-c} with more than negligible advantage over a coin toss (even though she learns $g(y_{1-c})$).

1.4 Our Results

Realizing that a combinatorial characterization does not suffice for categorizing all efficiently computable functions, one wonders whether at all there exist simple criteria that can capture the above notions in the general computational scenario? We answer this question positively.

This paper presents computational criteria for being in $\mathcal{SFE}$-C and Eff-$\mathcal{SFE}$. These criteria are computational in nature and hold for functions of unbounded input length. Also, they are very

[7] In one-to-one functions $f(x_0, y_0) \neq f(x_1, y_0)$ for all x_0, x_1, y_0. So no insecure minor can exist

[8] Under the existence of OT every function is complete for SFE. Nevertheless, the existence of one-to-one one-way functions seems a much weaker assumption and in particular does not imply OT with respect to black-box reductions [29].

close to being complementary to each other, thus almost fully categorizing all efficiently computable functions.

We define the following properties on a function f (the following are high level descriptions, for the full details see Definitions 4.1 and 3.1):

- f is called **(Computational) Row Transitive** if one can efficiently compute $f(x_1, y)$ when given x_0, x_1 and $f(x_0, y)$ for every x_0, x_1 and y.
- f is said to be **(Computational) Row Non-Transitive** if for some x_0, x_1 it is (somewhat) hard to compute $f(x_1, y)$ from x_0, x_1 and $f(x_0, y)$ for a random (unknown) y.

Comment: We call this property computational row transitivity since given a value of f taken from the row $f(x_0, \cdot)$ (viewing f as a table), Alice can efficiently deduce the corresponding value in row $f(x_1, \cdot)$. In essence this means that for Alice, learning the value at any row is equivalent.

Superficially, the two properties may seem exactly complementary to each other. However the exact formulations (Definitions 4.1 and 3.1) leaves a gap between the two.

Main Theorem *Let $f : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ be a polynomial time computable function.*

1. *If f is row transitive then it has an efficient SFE protocol ($f \in$ Eff-$\mathcal{SFE}$).*
2. *If f is row non-transitive then it is complete ($f \in \mathcal{SFE}$-C).*

Outline of Proof: In the following discussion we try to present the main ideas for the proofs, while the full proofs are presented in Sections 3 and 4. The proofs are conducted in the semi-honest model but the statements have implications in the malicious model as well (see Section 1.5).

(1) We show an efficient SFE protocol for a computational row transitive f. In general, the protocol simply consists of Bob choosing any input $\hat{x}$ and sending it to Alice along with $f(\hat{x}, y)$. Alice can then compute $f(x, y)$ by the row transitivity property. Also, the view of Alice can be simulated by choosing $\hat{x}$ and computing $f(\hat{x}, y)$ from her output $f(x, y)$ (again by the transitivity property).

(2) As for the row non-transitive functions, consider the following toy example: Suppose that we are given x_0, x_1 such that computing $f(x_1, y)$ from $f(x_0, y)$ is hard. Then using an SFE protocol for f we construct a version of OT we call Naive-OT, where the sender Bob has a secret bit b and the receiver Alice has a bit c that chooses whether she should learn Bob's secret or not (Bob learns nothing).[9] Let Bob choose y and Alice choose either x_0 or x_1. The two parties run the SFE protocol for f with their chosen inputs. Now Bob sends $b \oplus h(f(x_1, y))$, where h is the Goldreich-Levin (GL) hardcore bit ([20]). If Alice chose x_1, then she learned $f(x_1, y)$ in the SFE and hence can learn Bob's secret bit, however if she chose x_0 she only learned $f(x_0, y)$ and cannot predict the hard core bit with better than negligible probability. Bob learns essentially nothing in the process.

However, the actual proof of security is not as straightforward and calls for careful treatment: To begin with, we note that our use of the GL hardcore bit is somewhat non-standard as it is not applied to a one-way function.[10] Furthermore, the proof that the GL bit is hard to predict requires the hardness of computing $f(x_1, y)$ to be "strong hardness" (as is the case with one-way functions where strong one-way functions are needed), but our definition of row non-transitivity guarantees only "weak hardness" (this is crucial in order to avoid a big gap between the criteria for

[9] In the semi-honest model, Naive-OT is equivalent to the so called Rabin-OT, that in turn was shown to be equivalent to 1-2-OT [11]

[10] We assume that it is hard to compute $f(x_1, y)$ given $f(x_0, y)$, but have no guarantee that it is easy to compute $f(x_0, y)$ from $f(x_1, y)$, thus, this is a computation that might be hard in both ways.

$\mathcal{SFE}$-C and Eff-$\mathcal{SFE}$). To overcome this we show that the GL bit is only "weakly hard" to predict, and thus create an OT with weakened security (called Weak-OT[11] in this paper). We then show how one can use this weakened version of OT to construct a fully secure OT. This is shown by an amplification argument using Yao's XOR lemma.

Note that an implementation of OT was shown to guarantee secure computation for further computational models such as multi-party secure computation [21] (as opposed to two parties), symmetric versions of secure computation and secure computation of probabilistic functionalities. Thus the completeness theorem can be informally stated as: The existence of an SFE protocol for any row non-transitive function implies a wide range of multi party secure computation.

1.5 Applying the Results to the Malicious Model

Our paper and results concentrated thus far the semi-honest model (sometimes referred to as the "Honest but Curious" model). This means Alice and Bob follow the protocol exactly as specified (they are honest), but after the protocol is executed, the parties may try to extract more information than actually intended by inspecting the transcript of the protocol (they are curious). A major justification for this assumption stems the compiler of Goldreich, Micali and Wigderson [22], that showed how every protocol that is secure in the semi-honest model, can be transformed into a protocol secure in the malicious model (assuming (non-uniform) one-way functions exist). This means that assuming one-way functions do exist, the results presented carry over to the real world were parties may be malicious. More precisely:

Theorem 1.1 *If one-way functions exist then:*

- *All row-transitive functions have efficient SFE protocols secure against malicious parties.*
- *If a row non-transitive function has an efficient SFE protocol (even in the semi-honest model), then all functions have SFE protocols secure against malicious adversaries.*

It is important to point out that in the above statements the notion of SFE protocol must refer only to sending messages and tossing coins, without use of "magic boxes" such as third parties, noisy channels or quantum channels, where the GMW compiler does not apply.

Note that, on the other hand, the malicious and semi-honest worlds differ from one another under information theoretic security definitions, as implied by [2, 32] (see further discussion in Section 5).

Theorem 1.1 relies on the fair assumption that one-way functions exist. We further ask what can be said about the malicious model without this assumption? We point out that the existence an SFE for a complete function implies semi-honest OT that in turn implies the existence of one-way functions [28]. So the completeness part of the theorem holds unconditionally:

Theorem 1.2 *If a row non-transitive function has an efficient SFE protocol (even in the semi-honest model), then all polynomial time functions have SFE protocols secure against malicious adversaries.*

On the other hand, if one-way functions are not guaranteed, we do not know if there exists an SFE in the malicious model for a all row transitive functions. Moreover, we show a possible scenario (under certain plausible assumptions) in which there are functions that have SFE in the semi-honest model but not in the malicious model. This example along with the other statements raised in this subsection are thoroughly discussed in Section 5.

[11] The notion of Weak-OT in this paper (defined in Section 3) is different than various versions of OT with the same name defined in other papers (e.g. [13, 32])

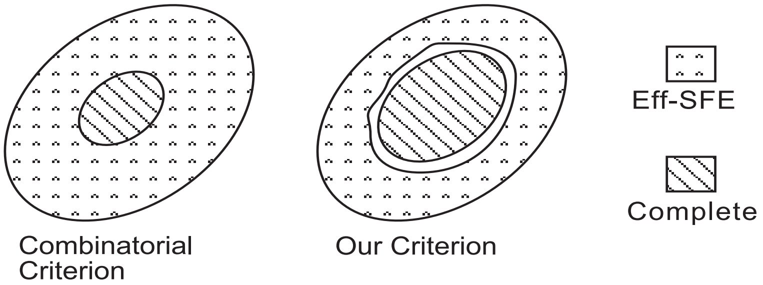

Figure 1: The area inside the outer circle depicts all efficiently computable functions. The left picture shows the tight characterization of functions according to the insecure minor criterion, while the right picture shows the characterization according to the "row transitivity" criteria.

1.6 The Meaning of our Result

The main theorem can be viewed as essentially categorizing all "nice" functions as either in $\mathcal{SFE}$-C or unconditionally in Eff-$\mathcal{SFE}$. Such results have been proved before, but were never shown for all functions on large inputs, where computational considerations come into account. This new view allows the inclusion of more functions in the family $\mathcal{SFE}$-C, that were considered as having "trivial" SFE by the combinatorial criterion of [2]. The price paid is that the characterization by row transitivity is not as tight as the one given by the insecure minor criterion, and leaves a gap between the criteria (as depicted in Figure 1). The functions in the gap, however, are functions with somewhat "unnatural" behavior such as functions that have hardness only on inputs from a distribution that is not samplable or functions that behave erratically on different input lengths (Section 6 contains a detailed discussion of this gap). Moreover, it seems that this gap cannot be closed altogether as implied by examples of functions tailored to be in neither category. Hence, this picture seems to reflect the actual state of affairs.

A Complexity Discussion: In order to introduce this work from a complexity point of view, we turn to the paper of Impagliazzo [27] that describes five possibile scenarios for the computational world according to different computational assumptions. Specifically, we mention the world 'Cryptomania' in which OT exists and the weaker world 'Minicrypt' where OT does *not* exist, but one-way functions do.

If in Cryptomania and OT protocols exist, then all efficiently computable functions are both complete and have SFE protocols ($\mathcal{SFE}$-C = Eff-$\mathcal{SFE}$). In such a world our result has no implication complexity wise, although it is still interesting as a tool for constructing OT (see Example 2 below). On the other hand, when considering Minicrypt, one can ask whether there are stronger assumptions than the existence of one-way functions that are meaningful. For example, is there an assumption that allows the secure evaluation of a family of non-trivial functions? Our results imply that the answer to the latter question is no. In other words, our main theorem essentially claims that as far as SFE protocols go, there are no additional worlds between Minicrypt and Cryptomania.

Possible Applications: The result also has interesting aspects even in the case that OT does exist. This is due to the constructive nature of the proof in the sense that it actually describes how to construct an OT from an SFE for $f \in \mathcal{SFE}$-C. Consider the following example:

Example 2 *Consider the function $f_p(g, y) = g^y \mod p$, where p is a prime and q is a prime dividing $p - 1$.*[12] *Let g be chosen from a multiplicative subgroup $Q \subseteq \mathbb{Z}_p^*$ of order q, and $y \in [q]$. Let the security parameter n be $|p|$.*

The above function has a simple SFE protocol where Alice knows g, Bob knows y and at the end of the protocol Alice learns $g^y \mod p$. The protocol goes as follows: Let Alice choose a random $r \in [q]$ and send g^r to Bob. Bob then computes $z = g^{ry} = (g^r)^y$ and sends it to Alice. Alice now takes the rth root from z and gets $z^{r^{-1}} = (g^{ry})^{r^{-1}} = g^y$. It is simple to show that the view of either side can be simulated and that this is indeed an SFE for f.

Furthermore, under the Computational Diffie Helman (CDH) assumption, f is computational row non-transitive (and therefore SFE-complete), since one cannot efficiently compute g_1^y given g_0, g_1 and g_0^y (on the other hand, note that this function does not contain an insecure minor).

Now running the reduction from the proof of our main theorem actually produces the following protocol for Naive-OT: Let Bob be the sender holding a secret bit b and Alice be the receiver holding a choice bit c. Alice chooses random generators g_0, g_1 and $r \in \mathbb{Z}_p \setminus \{0\}$ and sends g_0, g_1, g_c^r to Bob. Bob computes $z = g_c^{ry}$ and g_1^y and sends $z, h(g_1^y) \oplus b$ to Alice ($h(\cdot)$ again stands for the GL hardcore bit). Finally, Alice computes $z^{r^{-1}} = g_c^y$. If $c = 1$ Alice can compute $h(g_1^y)$ and subsequently learn the secret bit b. However, if $c = 0$ then Alice learns practically nothing.

This turns out to be equivalent to the well known OT protocol of Bellare and Micali [3]. It remains open whether one can use this framework to give new constructions for OT, using SFE protocols for other complete functions.

Another possible application of the row non-transitivity criterion is as a tool to simply prove that a function is SFE-complete.

Example 3 *Consider the function $f_N(x, y) = (x + y)^3 \mod N$, where $N = p \cdot q$ for large primes p and q (The factorization of N is unknown) such that the number 3 is relatively prime to both $p - 1$ and $q - 1$.*[13]

Notice that each row in the function f_N is a permutation and hence no insecure minor exists. So at first glance it is unclear if this function is SFE-complete or perhaps it has a simple SFE protocol? We argue that under the RSA assumption f_N is row non-transitive. The general idea is that if f_N is row transitive, then given $a = (z - 1)^3 \mod N$ one can compute the value $z - 1$, thus contradicting the hardness of RSA. This is done by computing $b = (z + 1)^3 \mod N$ from the value a (this is possible given the row transitivity). Then the value z can be found, since $a - b = 2z^2 + 2 \mod N$, giving the value of $z^2 \mod N$ and in turn $z = z^3/z^2 \mod N$. This only showed that if f_N is row transitive it contradicts the RSA assumption. A more careful analysis yields that this function is indeed row non-transitive under the RSA assumption.

1.7 Information Theoretic vs. Computational

In this work we emphasize the importance of taking computational consideration into account. We now compare the three main models of secure function evaluation considered in the literature.[14] Each of these models has its merit and captures some aspects of the issue, while being limited in others. The specification of a model consists of three entities: the *players* executing the protocol, the *simulator* for each party (such a simulator gets a party's input together with its output and generates the view the party sees in the protocol) and the *distinguisher* that attempts to guess

[12] For convenience choose p to be of the form $p = 2q + 1$ for prime q.

[13] Assume here that N is given by an external source.

[14] Note the three models as well as their names, represent the authors personal view of the main possible models.

whether what it receives is the real or simulated view.[15] The difference between the three models is in the computational power of each of the entities.

The Fully Information Theoretic Model: In this model there are no limitations on the power of all the related entities (even the parties running the protocols). This is a highly unrealistic model but can yet be useful, especially when trying to prove impossibility results. For example, in this model it is shown that there exist functions that *cannot* be securely evaluated under information theoretic definitions. This was indeed proved for various simple functions as the OR of two bits and oblivious transfer (OT) ([4, 10] and others). These impossibility results hold over to the more realistic model of Unbounded Adversaries (presented next).

This model is also applicable when implementing secure evaluation of functions with constant domain size or when discussing secure reductions between two such primitives (sometimes referred to as as "Cryptogates"). In such a case, computational considerations are not a factor due to the small input size.

Note that in this model, it is possible to talk about the SFE of any function f, even one that is hard to compute, unlike the other models (described next) where the parties are expected to be able to efficiently compute the function f when no security requirements are made.

The Unbounded Adversaries Model: In this model, the parties running the protocol are required to be efficient (the protocol has to run in polynomial time), furthermore, the simulators should also be efficient. The distinguishers, on the other hand, are unbounded. This is a very realistic setting: it asks that a real world party (that can only run probabilistic polynomial time procedures) could simulate the view of the protocol to such perfection that even an unbounded distinguisher would not be able to tell the simulated view apart from the real view. Hence, this catches the notion that real bounded parties don't gain any information at all from participating in the protocol. Indeed, this model is advocated throughout the literature (e.g. [8, 19, 25, 38]).

This formulation is also convenient in the sense that reductions in this model carry over automatically to the computational model, that is, if one can construct an SFE for a function f using calls to an SFE for a function g in the unbounded model (we call this f "securely reduces" to g, defined in Section 2.2), then g also "securely reduces" to f in the computational model.

This seems a very appealing definition to work with, but has a drawback since it is actually impossible to achieve secure computation of many interesting functions, as is implied by the results for the fully information theoretic model. That is, in this model, no function is both complete and has an SFE protocol (the sets $\mathcal{SFE}$-C and Eff-$\mathcal{SFE}$ do not intersect at all). The notion of completeness is therefore interpreted as follows: a complete function f is a function that cannot be securely evaluated, but if given a "magic box" that evaluates f, it can be used to securely evaluate all functions. In particular, the introduction of various natural implementations to such "magic boxes" yields the ability to achieve secure computation in various models such as noisy channels (e.g. [12]), quantum channels (e.g. [5]), the bounded storage model (e.g. [7]) and multi party computations (e.g. [4, 9]), see other examples in Maurer [37].

We note that when discussing functions on a domain of constant size, the unbounded adversary model is equivalent to the fully information theoretic model (as computational considerations do not matter given the small domain).

[15] We note that the roles of the parties, simulators and distinguishers as described above corresponds to the semi-honest model. However, these entities may generalized to apply to the malicious model as well.

The Computational Model: In the computational model, all the related entities are computationally bounded. That is the protocol is efficient, as well as all simulators and distinguishers. This is a relaxation of the previous model in the sense that the simulator has to fool only bounded adversaries. Indeed, in this model, under the plausible assumption that OT exists, SFE of every efficiently computable function may be achieved, making this a very interesting setting to work under.

Another upside of working in a computational setting is the applicability of the GMW compiler [22] that converts every protocol that is secure in the semi-honest model to one that is secure in the malicious model, under the assumption that one-way functions exist. This allows applying results from the semi-honest world to the malicious world (see Sections 1.5 and 5) .

Each of the above models illuminates some points and hides others. The choice of model depends on the goals that one sets. For example, if one can allow strong setup assumptions such as an honest majority or the existence of a trusted party, then one should use the unbounded adversaries model as it is both realistic and gives unconditional and strong security. When discussing reductions between "cryptogates", the powerful fully information theoretic model may be used. If, on the other hand, SFE via classical protocols (with no "magic") is needed, one must work in the computational model, as it is the only model that may achieve this task.

The Relation between the Models: Clearly, every unbounded adversaries protocol is also secure in the fully information theoretic model (since the unbounded model is just a more restricted model). Therefore an impossibility result in the fully information theoretic model also yields a similar result for the unbounded adversaries model. Furthermore, an SFE protocol in the unbounded adversaries model is also an SFE protocol in the computational model, since the computational model only adds limitations on the distinguisher.

Denote by IT-$\mathcal{SFE}$-C and IT-Eff-$\mathcal{SFE}$ are the sets of complete functions and those with efficient SFE protocols in the fully information theoretic model, UA-$\mathcal{SFE}$-C and UA-Eff-$\mathcal{SFE}$ these sets in the unbounded adversaries model and Co-$\mathcal{SFE}$-C and Co-Eff-$\mathcal{SFE}$ denote these sets in the computational model. As far as the semi-honest asymmetric SFE goes, we know the following:

- As mentioned above, every protocol secure in the unbounded adversaries model is also secure both in fully information theoretic model and in the computational model. Therefore we have that UA-Eff-$\mathcal{SFE} \subseteq$ IT-Eff-$\mathcal{SFE}$ and also UA-Eff-$\mathcal{SFE} \subseteq$ Co-Eff-$\mathcal{SFE}$.

- In the work of Beimel, Malkin and Micali [2] it is shown that IT-$\mathcal{SFE}$-C $\cap$ IT-Eff-$\mathcal{SFE} = \emptyset$ and also that all functions are in either IT-$\mathcal{SFE}$-C or IT-Eff-$\mathcal{SFE}$ (depending if they have an insecure minor or not[16]).

- Since the reduction from OT to a function with an insecure minor is efficient, we have that IT-$\mathcal{SFE}$-C = UA-$\mathcal{SFE}$-C and they include exactly all functions with an insecure minor. [17]

- In Example 1 we show a function that is in IT-Eff-$\mathcal{SFE}$ but on the other hand in Co-$\mathcal{SFE}$-C (under a computational assumption). Hence UA-$\mathcal{SFE}$-C $\subsetneq$ Co-$\mathcal{SFE}$-C under the assumption

[16] Insecure minor may be defined in various ways when allowing unbounded input length. For instance, in this statement it suffices for a function to have at least one input length with an insecure minor. In the computational setting, on the other hand, it is required to have infinitely many input lengths with an insecure minor.

[17] This disregards uniformity issues (such as how to find the insecure minor). When uniformity is taken into account we can only say that UA-$\mathcal{SFE}$-C $\subseteq$ IT-$\mathcal{SFE}$-C.

that one-way permutations exist.[18] This shows a separation between the computational model and the unbounded adversaries and information theoretic models.

- An interesting question is whether a similar separation can be found between the unbounded adversaries and the fully information theoretic model. That is, is there a function with no insecure minor that is not in UA-Eff-$\mathcal{SFE}$ under reasonable assumptions? In Appendix A We give a partial answer to this question. We show a separation in the malicious model, and some indication towards a separation in the semi-honest model as well.

1.8 Paper Organization

In Section 2 we present the relevant formal definitions, including the notions of reductions and completeness. In Sections 3 and 4 we present our criteria and prove the main theorems. Section 5 elaborates on the applicability of our results in the malicious adversaries model. Section 6 discusses the functions in the gap between the two criteria. Section 7 shows the amplification of Weak-OT to strong OT. Section 8 mentions some further issues and questions. In Appendix A we discuss separating the fully information theoretic model from the unbounded adversaries model.

2 Formal Setting

Some general notations for this paper:

- PPTM stands for Probabilistic Polynomial Time Turing Machine.
- By a **Distributions Ensemble** we mean a series $\{D_s\}_{s \in S}$ where S is an infinite set of strings and D_s is a distribution.
- A **Samplable Distribution** is a PPTM D accepting a unary number and outputting a binary string of polynomially related length i.e $D(1^n) \in \{0,1\}^{l(n)}$ where $l(n)$ is bounded by a polynomial. This corresponds to a distribution ensemble $\{D_n\}_{n \in \mathbb{N}}$ that can be efficiently sampled by a uniform Turing machine.
- We choose to define security in this paper to be against non-uniform adversaries. These are are presented as Turing Machines that receive also an auxiliary information string, a formulation equivalent to that of circuits. Formally: a **PPTMA** is a PPTM with auxiliary information, that is, a probabilistic Turing machine that is required to run in time polynomial in the length of its first input (usually this is the security parameter) and has access to an auxiliary information string $w \in \{0,1\}^*$.

  On the one hand, the choice of non-uniform adversaries is essential when working in the malicious adversaries model (it is necessary, for example, for Theorem 1.2). On the other hand, most results hold also with definitions of security against uniform adversaries (definitions that also leave a smaller gap between the criteria, see section 6 for more details).
- Let S be an unbounded set of strings and let $\{X_s\}_{s \in S}$ and $\{Y_s\}_{s \in S}$ be distribution ensembles. We say that $\{X_s\}$ and $\{Y_s\}$ are **computationally indistinguishable** (Denoted $\{X_s\} \overset{c}{\approx}$

[18] If non-uniformity is allowed, then the existence of one-way functions (not permutations) suffices to show the separation: Construct a row non-transitive function f using the hardness of a one-way function. To makes sure that no insecure minor exists in the function f, add to the output a perfectly binding commitment of the input. Such a commitment can be achieve non-uniformly given one-way functions.

$\{Y_s\}$) if for every PPTMA M of polynomial size circuits, every polynomial $q(\cdot)$, all sufficiently large n, all $s \in S \cap \{0,1\}^n$ and all auxiliary information $w \in \{0,1\}^*$ we have:

$$|\Pr[M(1^n, X_s, w) = 1] - \Pr[M(1^n, Y_s, w) = 1]| < \frac{1}{q(n)}$$

The probability is taken over the distributions X_s, X_y and the randomness of M.

2.1 Secure Function Evaluation

Discussing the various definitions of SFE is beyond the scope of this paper. The reader is referred to [19] for a good overview of possible definitions. In this work we focus on a specific version of SFE, namely the semi-honest computational asymmetric version, where only one party gets the output of the deterministic function f (as opposed to probabilistic functionality). The definitions we present here are along the lines of [19].

Unlike the discussion of functions of constant input size, we allow the functions to receive very long inputs. This is done in the standard fashion by relating the complexity and security parameters to the function's input length. In order to accommodate this convention, we make the following assumptions:

- f is computable in time polynomial in the input length (combined length of the inputs).
- The security parameter for the SFE protocol is the input length (usually denoted n).
- For sake of simplicity and without loss of generality, assume that for input length n the output is always of length $m(n)$ (where $m(\cdot)$ is a polynomial). This assumption is justified by a padding argument.[19]
- Assume that all parties know n (the length of the input). Also, when giving partial inputs to various PPTM, one should also give 1^n as an extra input. However we omit this extra parameter for ease of notation.

Let Π be a protocol between Alice and Bob. Denote Alice's output by $\Pi_A(x, y)$ and Bob's output by $\Pi_B(x, y)$. In our case we simply denote $\Pi(x, y) = \Pi_A(x, y)$ (since $\Pi_B(x, y)$ is always empty). Denote Alice's view of the protocol by $\text{VIEW}_A^\Pi(x, y)$ (this includes Alice's local input, local randomness, her output and all of the messages received from Bob). Similarly Bob's view is $\text{VIEW}_B^\Pi(x, y)$.

The formal definition of an SFE protocol requires that each party's view of the protocol can be efficiently simulated, even when only seeing the party's local input and output. Hence practically nothing is gained by seeing the view of the protocol.

Definition 2.1 (SFE Protocol) *Let $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be a poly-time function. A polynomial time protocol Π is a* **Secure Function Evaluation (SFE)** *for f if the following holds:*

1. *Correctness: For every $x, y \in \{0,1\}^*$: $\Pi(x, y) = f(x, y)$.*
2. *Security:*
   - *Bob's Privacy: There exists a PPTM S_A such that:*

$$\{S_A(x, f(x, y))\}_{x,y \in \{0,1\}^*} \overset{c}{\approx} \{\text{VIEW}_A^\Pi(x, y)\}_{x,y \in \{0,1\}^*}$$

[19] Any general function can be turned into such a function - If the output is too short, we simply pad it with zeros to the appropriate length.

- *Alice's Privacy: There exists a PPTM S_B such that:*

$$\{S_B(y)\}_{x,y\in\{0,1\}^*} \overset{c}{\approx} \{\text{VIEW}^{\Pi}_B(x,y)\}_{x,y\in\{0,1\}^*}$$

Denote the set of functions that have an SFE protocol by Eff-$\mathcal{SFE}$.

We stress that the above definition (adapted from [19], Definition 7.2.1) applies only in the semi-honest model. The definition for the case of malicious parties introduces an "ideal box" that evaluates f, where both parties enter their respective inputs and Alice receives the output. The definition asks that in the real computation the parties gain nothing over calling an ideal box, in the sense that the view of the real computation can be simulated given only the view of the ideal box. This alternative approach can capture both the malicious and semi-honest models. For the exact definitions we refer the reader to [19, 8].

2.2 Reductions and Completeness in SFE

The definitions of reductions and completeness in the context of SFE are simply analogues of these notions in general (polynomial time) computation. We say that g securely reduces to f if one can construct an SFE protocol for g using calls to an "ideal box" that evaluates a function f. A function f is SFE-complete if given an SFE protocol for f, it is possible to securely evaluate any function. Formally:

- An *ideal box evaluating a function* f is a box that takes two inputs, x from Alice and y from Bob, and outputs $f(x,y)$ to Alice.[20]
- A *protocol with access to an ideal box* is a protocol where the two parties are allowed to flip coins, exchange messages and jointly invoke calls to an ideal box evaluating some function f. This means that the two parties can compute local inputs for f (one acting as Alice and the other as Bob) and send their respective inputs to the ideal box. The ideal box , in turn, returns its output to the party acting as Alice.
- A protocol with access to an ideal box is said to be an *SFE of a function* g if it follows definition 2.1 for SFE (only here the views also incorporate each of the parties' local view of the ideal box calls).

Definition 2.2 (Secure Reduction) *We say that a polynomial time function $g : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$* **securely reduces** *to a polynomial time function $f : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ if there exists a polynomial time protocol Π^f_g (polynomial in the length of the input to g) with access to an ideal box for evaluating f such that the protocol Π^f_g is an SFE for g.*

Definition 2.3 (SFE-Complete) *We say that a function f is* **SFE-Complete** *if every efficiently computable function g securely reduces to f. Denote:*

$$\mathcal{SFE}\text{-}C = \{f |\ f \textit{ is complete for SFE }\}$$

These definitions have the following essential composability properties (as shown by Canetti [8]):

- If g securely reduces to f and f has an SFE protocol then g has an SFE protocol.
- If g securely reduces to f and g is SFE-complete then f is SFE-complete.

[20] Bob does not see Alice's interaction with the ideal box and vice versa.

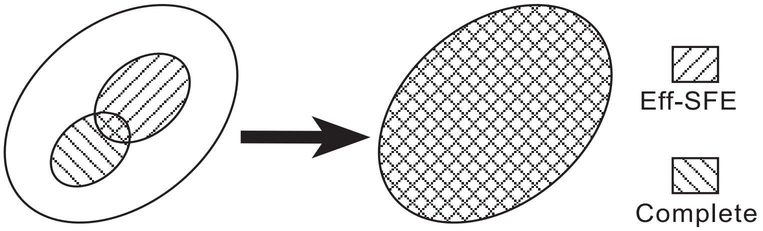

Figure 2: The two families $\mathcal{SFE}$-C and Eff-$\mathcal{SFE}$ may have no intersection, but if they intersect (as is shown in the figure) then all efficiently computable functions have SFE protocols and also $\mathcal{SFE}$-C $=$ Eff-$\mathcal{SFE}$.

The sets $\mathcal{SFE}$-C and Eff-$\mathcal{SFE}$ have the following two possibilities: Either the two sets are distinct (have no intersection) or the two sets have an intersection. But the latter case implies that all efficiently computable functions have an SFE protocol, and thus $\mathcal{SFE}$-C $=$ Eff-$\mathcal{SFE}$ and they contain all efficiently computable functions (see Figure 2). This is indeed the case, for instance, if efficient OT protocols (defined in the next section) exist. Altogether, either $\mathcal{SFE}$-C and Eff-$\mathcal{SFE}$ do not intersect or $\mathcal{SFE}$-C $=$ Eff-$\mathcal{SFE}$ and they contain all efficiently computable functions.

Note that this is also the situation for other definitions of completeness such as $\mathcal{NP}$-completeness, but the SFE scenario differs that of $\mathcal{NP}$ in the sense that for $\mathcal{NP}$ it is widely assumed that $\mathcal{P} \neq \mathcal{NP}$ (and therefore the sets $\mathcal{NP}$-C and $\mathcal{P}$ are assumed to be distinct). While in our case of SFE it is seems reasonable (or at least not surprising) that there is an intersection and that all functions are both SFE-complete and in Eff-$\mathcal{SFE}$. Perhaps a better analogue to the situation in SFE-completeness is in the world of logspace computation, where the actual situation is not clear, either $\mathcal{NL} = \mathcal{L}$ or all logspace complete functions are not in L ($\mathcal{NL}$-C $\cap\, \mathcal{L} = \emptyset$).

This paper aims at finding a simple criterion as to what functions are in $\mathcal{SFE}$-C. The common method of proving that a function is SFE-complete is showing a secure reduction from a previously known complete function, namely, from Oblivious Transfer.

2.3 Oblivious Transfer

A central component in the construction of SFE protocols is the Oblivious Transfer (OT) protocol. OT refers to several equivalent versions of two-party protocols. For example, an important formulation is the 1-2-OT (due to [14]), where Bob has two secret bits b_0, b_1 and Alice has a choice bit c. At the end of the protocol Alice learns b_c but learns nothing about b_{1-c}, while Bob learns nothing about Alice's choice. In principal, we can view a 1-2-OT protocol through the framework of SFE. Namely, an SFE protocol for the function $f(c,(b_0,b_1)) = b_c$.

Another important version is known as Noisy-OT or Rabin-OT (due to Rabin [42]), and was shown to be equivalent to 1-2-OT in [11]. This version goes as follows: Bob holds a secret bit b. After the protocol, Alice receives the bit b with probability $\frac{1}{2}$ (with probability $\frac{1}{2}$ she learns nothing), and Bob doesn't know if Alice received the bit or not.

In this paper we use a slightly different version that we call Naive-OT, that is clearly equivalent to the Rabin-OT in the semi-honest model. In this version Alice simply chooses whether to receive Bob's secret bit or not, while Bob learns nothing of this choice.

Definition 2.4 *A* **Naive-OT** *protocol is an SFE protocol for the function:*

$$f(c,b) = \begin{cases} b & c = 1 \\ \perp & c = 0 \end{cases}$$

As mentioned before, the importance of OT stems from the fact that it is complete for SFE. This was shown in an array of works, originally attributed to [43, 45] with a line of further works for different models of secure computation, e.g. [21, 24, 30]. In turn, OT can be constructed from trapdoor permutations and public key encryption with particular "niceness" properties [14, 19, 17] and various specific intractability assumptions (e.g. the Diffie Helman assumptions [3, 39]). In contrast, OT cannot be reduced in a black-box manner to weaker primitives such as one-way functions, general public key encryption or even trapdoor one-to-one functions [29, 27, 17].

3 Completeness

Our main result presents criteria for functions to being complete or in Eff-$\mathcal{SFE}$. While the two criteria are not complementary of each other, they are in a sense a "strong negation" of each other. For simplicity of exposition, the criteria we present in Sections 3 and 4 are not as tight as possible, and give results for the most natural definition of SFE. In Section 6 we discuss alternative definitions that bring us closer to a tight characterization.

Definition 3.1 (Computational Row Non-Transitivity:) *We say that a function $f : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ is* **(Computational) Row Non-Transitive** *if there exist polynomial-time samplable distributions $D_x(\cdot), D_y(\cdot)$ and there exists a polynomial $p(\cdot)$ such that for every PPTMA M, for all auxiliary information $w \in \{0,1\}^*$ and for all but finitely many n:*

$$\Pr[M(x_0, x_1, f(x_0, y), w) = f(x_1, y)] < 1 - \frac{1}{p(n)}$$

Where the probability is taken over $x_0, x_1 \in D_x(1^n), y \in D_y(1^n)$ and the randomness of M.[21]

We note that the notion of hardness in row non-transitivity is very different than that of a one-way function. To begin with, row non-transitivity guarantees that it is hard to compute $f(x_1, y)$ given $f(x_0, y)$, but gives no guarantee that it is easy to compute $f(x_0, y)$ from $f(x_1, y)$, thus, this is a computation that might be hard in both ways (even when x_0 and x_1 are fixed values and not random variables). Furthermore, unlike one-way functions where one is given $g(y)$ and asked to find any z such that $g(z) = g(y)$, here we ask only that it is impossible to find the specific value $f(x_1, y)$ from $f(x_0, y)$, while putting no restriction on the ability of finding a value $f(x_1, z)$ with $f(x_0, z) = f(x_0, y)$. As a result, row non-transitivity does not even imply the existence of one-way functions, and its hardness can come also from a non computational source (for example the OR of two bits is a row non-transitive function).

Theorem 3.1 *If a function f is computational row non-transitive then it is complete for SFE.*

To prove the theorem we show a construction of an OT protocol using access to an ideal box for evaluating a non-transitive f. Rather than directly showing this, we first construct a weakened implementation of Naive-OT that is later shown to imply OT. This is called Weak-OT and is essentially a relaxation of the security restrictions on the receiver.

Definition 3.2 *A* **Weak implementation of Naive-OT (Weak-OT in short)** *consists of two parties, the sender and the receiver. The sender holds a secret bit b and the receiver holds a choice bit c. Both parties have a security parameter n. At the end of the protocol, the following holds:*

1. *Correctness: If $c = 1$ then the receiver outputs bit b.*

[21] Recall that a PPTMA is a PPTM with auxiliary information, that is polynomial time in the security parameter

2. Security:

- *Receiver's view: If $c = 0$ then there exists a polynomial $p'(\cdot)$ such that for every PPTMA A, for all auxiliary information $w \in \{0,1\}^*$ and for all but finitely many n:*

$$\Pr[A(\text{VIEW}^{weak-OT}_{receiver}(c,b), w) = b] < 1 - \frac{1}{p'(n)}$$

- *Sender's view: for every PPTMA B, for all polynomials $q(\cdot)$, for all auxiliary information $w \in \{0,1\}^*$ and for all but finitely many n:*

$$\Pr[B(\text{VIEW}^{weak-OT}_{sender}(c,b), w) = c] < \frac{1}{2} + \frac{1}{q(n)}$$

The definition of Weak-OT is justified by the following claim:

Lemma 3.2 *The existence of Weak-OT implies the existence of OT.*

The Lemma is proved in Section 7. We now turn to prove Theorem 3.1:

Proof. (of Theorem 3.1) We show a construction of a Weak-OT protocol using access to an ideal box for evaluating a row non-transitive f.

In the Weak-OT the sender holds a secret bit b and the receiver holds a choice bit c. The two sides call a box Π_f for the non-transitive function f. Let the receiver play Alice (holds x) and the sender play Bob (holds y).

Weak-OT$^{\Pi_f}(c, b)$:

1. The sender chooses random x_0, x_1 according to the distribution $D_x(1^n)$ and sends them to the receiver.
2. The receiver and sender jointly call the ideal box Π_f.

   The receiver uses $x = x_c$

   The sender chooses a random y according to $D_y(1^n)$.

   The receiver learns $f(x, y)$.
3. The sender:

   Computes $z = f(x_1, y)$ and chooses a string r uniformly at random in $\{0,1\}^{|z|}$.

   Sends to the receiver r and $\langle z, r\rangle \oplus b$, where $\langle z, r\rangle$ denotes the inner product of the strings z and r mod 2 (in other words this is the Goldreich-Levin predicate).
4. If $c = 1$ then the receiver retrieves the bit b by computing $\langle f(x, y), r\rangle = \langle z, r\rangle$.

We show that the above protocol constitutes of a Weak-OT protocol:

Correctness: If $c = 1$, then the receiver learns $f(x, y) = f(x_1, y) = z$, and can therefore learn $\langle z, r\rangle$ and can retrieve the bit b.

Security:

- Sender's view: The sender's view consists of his own messages as he only sends messages. In the access to the ideal box he only sees his input and does not see the output. He therefore has no information at all regarding the bit c.
- Receiver's view: If $c = 0$, then the receiver learns $f(x, y) = f(x_0, y)$. We show that the receiver cannot use this information to predict the bit $\langle f(x_1, y), r\rangle$ (and equivalently the bit b), with probability better than $1 - \frac{1}{p(n)}$ for some polynomial $p(\cdot)$. This is equivalent the weak version of Goldreich-Levin's hardcore bit where it is only required to show that the bit is somewhat hard to predict. The statement is formalized and proved in the following lemma.

Lemma 3.3 *Suppose f is row non-transitive. Then there exists a polynomial $p(\cdot)$ such that for all PPTMA A, for all auxiliary information $w \in \{0,1\}^*$ and for all but finitely many n's:*

$$\Pr[A(x_0, x_1, f(x_0, y), r, w) = \langle f(x_1, y), r\rangle] < 1 - \frac{1}{p(n)}$$

Where the probability is taken over $x_0, x_1 \in D_x(1^n), y \in D_y(1^n), r \in \{0,1\}^{|f(x_1,y)|}$ and the randomness of A.

Proof. Suppose for the sake of contradiction that there exists a PPTMA A that manages to predict $\langle f(x_1, y), r\rangle$ given $f(x_0, y)$ and r with overwhelming probability. More precisely: there exists a PPTMA A with auxiliary information $w \in \{0,1\}^*$ such that for all polynomials $q(\cdot)$, for infinitely many n's:

$$\Pr[A(x_0, x_1, f(x_0, y), r, w) = \langle f(x_1, y), r\rangle] > 1 - \frac{1}{q(n)} \tag{1}$$

We derive a contradiction to the non-transitivity of f by presenting a procedure B (with black box access to A) that correctly computes $f(x_1, y)$ given $f(x_0, y)$ with very high probability.

We concentrate our efforts on inputs y on which the algorithm A succeeds with probability greater than $\frac{9}{10}$. Define:

$$E_n = \{y|\Pr[A(x_0, x_1, f(x_0, y), r, w) = \langle f(x_1, y), r\rangle] > \frac{9}{10}\}$$

Since A has a very high success probability, the set E_n must contain almost all of the inputs. This is formalized by the following easy claim (brought here without a proof):

Claim 3.4 *For all polynomials $q(\cdot)$, for all but finitely many n's:*

$$\Pr_{y \in D_y(1^n)}[E_n] > 1 - \frac{1}{q(n)}$$

The claim is specifically true for $q(n) = 2p(n)$, where $p(\cdot)$ is the polynomial given by the non-transitivity of f. So $\Pr_{y \in D_y(1^n)}[E_n] > 1 - \frac{1}{2p(n)}$ for all but finitely many n.

Next we introduce the procedure B that uses A to compute $f(x_1, y)$ given $f(x_0, y)$ for all $y \in E_n$ and with very high probability (over the procedure's random bits). Consider a specific $y \in E_n$. The computation of $f(x_1, y)$ is done bit by bit. In order to compute the ith bit (denoted $f(x_1, y)_i$), choose a random $r \in \{0,1\}^{|f(x_1,y)|}$ and compute $A(x_0, x_1, f(x_0, y), r, w)$ and $A(x_0, x_1, f(x_0, y), r \oplus e^i, w)$

(where e^i is the binary vector with one in the ith place and zero in all others). If A succeeds on both inputs then:

$$\begin{aligned} A(x_0, x_1, f(x_0, y), r, w) \oplus A(x_0, x_1, f(x_0, y), r \oplus e^i, w) &= \langle f(x_1, y), r\rangle \oplus \langle f(x_1, y), r \oplus e^i\rangle \\ &= f(x_1, y)_i \end{aligned}$$

The probability that A succeeds on both of the above inputs is at least $1 - 2 \cdot (1 - \frac{9}{10}) = \frac{8}{10}$. Repeating this n times and taking a majority gives the bit $f(x_1, y)_i$ with exponentially small error probability $2^{-\Omega(n)}$ (by a Chernoff bound). This procedure is carried out for every bit of $f(x_1, y)$ separately, ultimately outputting the full string $f(x_1, y)$ with probability at least $1 - n \cdot 2^{-\Omega(n)} = 1 - 2^{-\Omega(n)}$ for infinitely many n. Combining this with Claim 3.4 and looking at all inputs $y \in D_y(1^n)$, the described efficient procedure B computes $f(x_1, y)$ from $f(x_0, y)$ with probability at least: $\Pr[E_n] \cdot (1 - 2^{-\Omega(n)}) > (1 - \frac{1}{2p(n)})(1 - 2^{-\Omega(n)}) > 1 - \frac{1}{p(n)}$, contradicting the non-transitivity of f.

□

To conclude the proof suppose that there exists a PPTMA A that predicts the bit b from the receiver's view, with probability better than $1 - \frac{1}{q(n)}$ for all polynomials $q(\cdot)$ and for infinitely many n's. The receiver's view only consists of $x_0, x_1, f(x_0, y)$ and the bit $\langle f(x_1, y), r\rangle \oplus b$. Thus A predicts $\langle f(x_1, r)\rangle$ given only $x_0, x_1, f(x_0, y)$ with probability better than $1 - \frac{1}{q(n)}$ for all polynomials $q(\cdot)$, and this contradicts Lemma 3.3. ■

Note that the approach we take in our proof of Theorem 3.1 is of taking a hardcore bit of a function that has only "weak" hardness, and later amplifying the hardness by repetition and XORing of the resulting weak hardcore bits (as shown in Section 7). An alternative approach would be to first amplify the hardness guaranteed by the row non-transitivity of the function, by considering a concatenation of many independent copies of the function f, and only then applying the GL hardcore bit. We note that the two suggested methods are equivalent.

Why use the Goldreich-Levin Hardcore Bit? Our choice of the GL hardcore bit stems from its generality, that is the fact that it applies to any computation that is hard. This is crucial since we know nothing in advance about the specific computation at hand (i.e. the computation of $f(x_1, y)$ given x_0, x_1 and $f(x_0, y)$). In fact, it is sufficient to have any choice of a function $h(x, r)$ with the following property: There exists a polynomial $p(\cdot)$ such that the input x can be retrieved efficiently with overwhelming probability (allowing only a negligible error) when given access to an oracle that outputs $h(x, r)$ correctly with probability at least $1 - \frac{1}{p(n)}$ for a random r. Interestingly, we can even use the function $h(x, i) = x_i$ which is very mildly hard. We prefer to use the GL bit as it implies a much more efficient construction for OT.

The BMM Criterion Implies Non-Transitivity An important test case for our completeness criterion is whether it encapsulates previous results, namely, the insecure minor criterion of [2]. Recall that an insecure minor consists of inputs x_0, x_1 and y_0, y_1 such that $f(x_0, y_0) = f(x_0, y_1)$ but $f(x_1, y_0) \neq f(x_1, y_1)$.[22]

Choosing the rows x_0, x_1 and the distribution D_y to be uniform on $\{y_0, y_1\}$ we get that for all PPTMA M (or any function at all in this case):

$$\Pr[M(x_0, x_1, f(x_0, y), w) = f(x_1, y)] \leq \frac{1}{2}$$

[22] We note that this definition is for functions of constant input length. The generalization to functions on unbounded input length requires to have an insecure minor for every input length n, and also requires that finding the values x_0, x_1 and y_0, y_1 can be done in polynomial time.

Hence the function f is also computational row non-transitive.

4 Criterion for Eff-$\mathcal{SFE}$

We now present our criterion for a function to be unconditionally in Eff-$\mathcal{SFE}$. This criterion is complementary in nature to the criterion for completeness, in the sense that while the computational row non-transitive condition requests the "weak hardness" of computing $f(x_1, y)$ from $f(x_0, y)$, the row transitivity condition requests that this computation is easy.

Definition 4.1 Computational Row Transitivity *We say that a function* $f : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ *is* **(computational) row transitive** *if there exists a PPTM M such that for all inputs* $x_0, x_1 \in \{0,1\}^n, y \in \{0,1\}^{n'}$ *:*

$$M(x_0, x_1, f(x_0, y)) = f(x_1, y)$$

We emphasis that in this definition M is required to be a uniform Turing Machine, as it will actually be used by the parties in the SFE protocol for f.

Theorem 4.1 *Let f be an efficiently computable function and suppose it is computational row transitive then f is unconditionally in Eff-$\mathcal{SFE}$.*[23]

Proof. Suppose that f is row transitive, then the following is an SFE protocol for f:

$\Pi_f(x, y)$

1. Bob chooses an arbitrary $\hat{x} \in \{0,1\}^n$ and sends $\hat{x}$ and $f(\hat{x}, y)$ to Alice.
2. Alice computes and outputs $M(\hat{x}, x, f(\hat{x}, y))$.

The above preocedure is easily shown to be an SFE protocol for f. The correctness follows since Alice learns $M(\hat{x}, x, f(\hat{x}, y)) = f(x, y)$ as required. As for security: Bob does not learn anything from the protocol simply because he gets no message from Alice. On the other hand, Alice's view can be simulated given the output $f(x, y)$, simply by choosing a random $\hat{x}$ and computing $M(x, \hat{x}, f(x, y)) = f(\hat{x}, y)$. ■

5 The Semi-Honest vs. the Malicious Model

Throughout the paper, we allow ourselves to assume that Alice and Bob are semi-honest. This means that they follow the protocol exactly as specified, but after the protocol is executed, the parties may try to extract more information than actually intended by inspecting the transcript of the protocol. Working in this model can be viewed as a stepping stone towards achieving security in the more realistic malicious model where the parties can run any desirable strategy and may choose to violate the prescribed protocol. The next step is transforming semi-honest protocols into malicious ones. This can be achieved using the compiler of Goldreich, Micali and Wigderson [22], that takes a protocol that is secure in the semi-honest model, and transforms it into a protocol secure in the malicious model (assuming (non-uniform) one-way functions exist). This means that assuming one-way functions do exist, the main theorem of this paper carries over to the malicious model:

[23] Note that Theorem 4.1 holds unconditionally, that is, the validity of the SFE protocol does not rely on any assumption (such as the existence of OT).

Theorem 5.1 *If (non-uniform) one-way functions exist then:*

- *All row-transitive functions have efficient SFE protocols secure against malicious parties.*
- *If a row non-transitive function has an efficient SFE protocol (even in the semi-honest model), then all functions have SFE protocols secure against malicious adversaries.*

Note that in the above theorem, "SFE protocol" refers only to sending messages and tossing coins, without use of "magic boxes" such as third parties, noisy channels or quantum channels, where the GMW compiler does not apply.

Row Non-Transitive functions in the Malicious Model The above theorem relies on the existence of one-way functions. We further show that the existence of SFE for any complete function implies the existence of one-way functions. Thus we get the following unconditional Theorem for row non-transitive functions:

Theorem 5.2 *If a row non-transitive function has an efficient SFE protocol (even in the semi-honest model), then all polynomial time functions have SFE protocols secure against malicious adversaries.*

Proof. Suppose that there exists an SFE protocol (in the semi-honest model) for a row non-transitive function f. Then by Theorem 3.1 there exists a construction of an OT protocol in the semi-honest model. By the result of Impagliazzo and Luby [28], the existence of semi-honest OT implies the existence of one-way functions.[24] Now, in order to construct a malicious SFE protocol for any efficiently computable function g, first construct a semi-honest SFE protocol for g using the reduction to OT [43], and given that one-way functions exist, run the GMW compiler on the semi-honest protocol for g to receive a malicious protocol for g. ■

Row Transitive functions in the Malicious Model On the other hand, if one-way functions are not guaranteed, we do not know if there exists an SFE in the malicious model for all row transitive functions. For example, in the above mentioned trivial SFE protocol for row transitive functions, Alice *cannot* act maliciously as she does not send any messages to Bob. However, it might be the case that a malicious Bob cheats by sending Alice an illegal value. If it is hard to distinguish a legal message from Bob from an illegal message, then such a protocol is no longer secure.

Moreover, we show that under certain assumptions (that are plausible within our current state of knowledge), there are functions that have SFE in the semi-honest model but not in the malicious model. More precisely, define the following weak notion of one-way functions:

Definition 5.1 *A collection of functions $\{f_i : D_i \to \{0,1\}^*\}_{i \in \bar{I}}$ has* **one-way instances** *if:*

1. *Easy to sample and compute: Have PPTMs to sample $i \in \bar{I}$ sample $x \in D_i$ and to compute $f_i(x)$.*
2. *Some functions are hard to invert: For very PPTM A, every polynomial $p(\cdot)$ and infinitely many $i \in \bar{I}$,*
$$\Pr[A(i, f_i(x)) \in f_i^{-1} f_i(x)] < \frac{1}{p(|i|)}$$
where probability is taken over $x \in D_i$ and the randomness of A.

[24] Actually, in [28] it was only shown that several other primitives (not OT) imply one-way functions, for example, this was shown for bit commitment. However, it is standard to show that OT implies bit commitment and hence also implies one-way functions

This weak notion of one-wayness was defined in [18] as part of the presentation of [41]. We can now state the following claim:

Claim 5.3 *Suppose that there exist no collections of functions with one-way instances and also* $\mathcal{NP} \nsubseteq \mathcal{P}/\mathrm{poly}$ *, then there exist functions* f *that have semi-honest SFE but no malicious SFE.*

Proof. A Theorem of Ostrovsky and Wigderson [41] states that if there exist no collections of functions with one-way instances then $\mathcal{ZK} = \mathcal{BPP}$, that is, the languages that have zero knowledge proofs are exactly those languages that are in $\mathcal{BPP}$.

Consider for example the problem of finding a Hamiltonian Cycle (HC) in a graph. Define $G(C, E)$ to be a graph with Hamiltonian cycle C on top of a basic edge set E. Define the function $f(x, y) = G(y)$, where $y = (C, E)$. The function f is definitely a row transitive function as it disregards the input x is identical for all rows, and the simple semi-honest SFE protocol will simply have Bob send the value $G(y)$ to Alice. In the malicious world it is requested, among other things, that Alice outputs a value $G(y)$ for some legal input y. Without loss of generality, Bob can start the protocol by sending $z = G(y)$ to Alice (since by the security of the SFE, Bob is required to choose y in advance and also Alice is bound to learn this value anyway by the end of the protocol). The rest of the protocol can be viewed as a zero knowledge proof that there exists a y such that $z = G(y)$. This is because by the properties of SFE, Alice should reject a z that is not in the image of G with high probability (providing the soundness requirement in ZK-proofs). Furthermore, the SFE definition ensures that the conversation can be simulated seeing only Alice's input x and output z, thus capturing the simulation requirement of zero knowledge. So, a malicious SFE for f is a zero knowledge proof to the existence of a hamiltonian cycle in a graph. Assume to the contrary that such a malicious SFE protocol for f exists, thus there is a zero knowledge proof to the existence of a hamiltonian cycle. But since hamiltonian cycle is an NP-complete problem, and we assumed that $\mathcal{NP} \nsubseteq \mathcal{P}/\mathrm{poly}$, then hamiltonian cycle is not in $\mathcal{BPP}$ and we get a contradiction. Altogether, under the assumptions above, the function f has a semi-honest SFE but cannot have a malicious SFE.

■

Semi-Honest vs. Malicious in the Information Theoretic model Finally we note that under information theoretic security definitions there is a difference between the malicious and semi-honest worlds (even under various assumptions). Indeed, Kilian [32] showed that the criterion for completeness in the malicious model is different than the insecure minor criterion of the semi-honest case (under such definitions the GMW compiler cannot apply as it promises only computational security). For example the OR of two bits is complete in the semi-honest case, but not so in the malicious model.

6 The Gap between the Criteria

Our ultimate goal is to categorize all functions as either in $\mathcal{SFE}$-C or in Eff-$\mathcal{SFE}$, however, our criteria fall short of this task and leave a gap of uncategorized functions. In this section we discuss this gap and what functions it contains, starting with the following important observations:

1. Some of the functions in the gap can in fact be categorized by giving more accurate or more relaxed definitions of the row transitivity criteria and the definitions of SFE. We avoided doing this before in order to present simple and clean definitions to go along with standard definitions of SFE.

2. The fact that a gap exists seem to reflect the actual world as there are functions that seem to be neither complete nor in Eff-$\mathcal{SFE}$. This state is very common in computational settings, and in fact, this characterization may be considered very tight as far as computational characterizations go.

3. The functions in the gap posses some "unnatural" behavior (see below) and thus we say that effectively the criteria cover all "nice" functions.

That being said, we elaborate on what are the types of functions that might be in the gap and how these functions may or may not be categorized after all:

- Functions for which it is possible to compute the $f(x_1, y)$ given x_0, x_1 and $f(x_0, y)$ by a non-uniform adversary (a circuit family) but hard to do so for a uniform adversary (PPTM):

  It is possible to define SFE to be secure only against uniform adversaries if working in the semi-honest model.[25] In such a case, row non-transitivity may be defined as having hardness for uniform machines, the main theorem will still hold, and the functions mentioned will be categorized as complete (in the semi-honest model). However, in order to apply the GMW transformation one must use one-way functions that are hard for non-uniform adversaries, and thus Theorem 1.2 does not hold anymore.

- Functions for which it is possible to compute the $f(x_1, y)$ given x_0, x_1 and $f(x_0, y)$ with overwhelming probability when x_0, x_1 and y are taken from any samplable distribution but hard to do so for inputs taken from some distribution that is not efficiently samplable: Considering that the inputs of an SFE should be taken from a samplable distribution, then these functions have an SFE protocol with the exception that the protocol can fail with negligible error.

- Functions that behave inconsistently on different input lengths:

  On the one hand, there are functions with such behavior that seem to be in neither category. For instance a function that alternates between a very trivial function to an inherently complete one (like OT) depending on the input length. If the occurrences of OT are sparse enough then the function cannot be complete, but still will not be in Eff-$\mathcal{SFE}$.

  On the other hand, there are functions in this category that are definitely complete. For instance, the requirement for "all but finitely many n" in the definition of row non-transitivity is much stronger than what is needed. The real need is that for any security parameter n we can efficiently choose an input length $l(n)$ (polynomially related to n) for which computing $f(x_1, y)$ from $f(x_0, y)$ is "weakly hard" (probability here should also include the randomness in choosing $l(n)$).

- Functions such that for every polynomial $q(\cdot)$ there exists a PPTM M_q that computes $f(x_1, y)$ given x_0, x_1 and $f(x_0, y)$ with success probability at least $1 - \frac{1}{q(n)}$, but there is no one PPTM M that achieves this for all possible q: Here it is unclear what can be done unless considering strong relaxations of the definitions of SFE.

7 Weak-OT implies OT

This section shows that the weak implementation of OT (Weak-OT) implies a strong OT. We note that this amplification result defers from previous such results like [12, 13] that were all information

[25] This is since composition theorems hold for uniform machines only under the semi-honest model

theoretic. Our amplification works against computational adversaries and requires more complex proofs using tools such as Yao's XOR-lemma.

Recall that the definition of a "strong" Naive-OT protocol requires that the views of the sender and receiver may be simulated (up to a negligible deviance), when seeing just their local input and output. This is equivalent to saying that the sender (or receiver) cannot guess the other side's input bit with non-negligible advantage over flipping a coin (when seeing their local view of the protocol). A Weak-OT is a relaxation of the latter definition. The sender is restricted in the same manner, but the receiver is allowed a higher success probability. We require that the receiver's success probability in guessing the sender's bit b is bounded by $1 - \frac{1}{p(n)}$ for a specific polynomial $p(\cdot)$. We show the following:

Lemma 3.2 *Weak-OT exists if and only if OT exists.*

Proof. Any OT protocol is also a Weak-OT protocol since it withstands harder security requirements. To prove that Weak-OT implies the existence of a strong Naive-OT, we present a construction of an OT protocol based on a protocol for Weak-OT. Let $p(\cdot)$ be the promised polynomial in the Weak-OT's definition of security for the receiver's view. Define $t(n) = p(n)^2$.

$\mathbf{OT}^{Weak-OT}(c, b)$**:**

1. The Sender randomly chooses $t(n)$ random bits $b_1, ..., b_{t(n)-1} \in_R \{0,1\}$ and sets $b_{t(n)}$ such that $b = \bigoplus_{i=1}^{t(n)} b_i$.
2. The sender and receiver run the protocol Weak-OT(c, b_i) for every i.
3. If $c = 0$, then the receiver computes $b = \bigoplus_{i=1}^{t(n)} b_i$.

Note: Stage 2 can be run in parallel since we are in the semi-honest model.

The correctness of the above protocol follows immediately.

The Sender's view: Suppose the sender can guess the bit c with advantage of $\frac{1}{q(n)}$ for a polynomial $q(\cdot)$. Then by a hybrid argument, there is an algorithm that can guess c on a single weak-OT run with advantage of $\frac{1}{t(n)q(n)}$ thus contradicting the weak-OT definition. So the sender can only guess the bit c with negligible advantage.

The Receiver's view: To show that the receiver's advantage is only negligible we apply the so called Yao's XOR-Lemma. Originally due to Yao (in presentations of [44]), with formal proofs presented in [36, 26, 23]. The XOR-Lemma states the following:

Suppose that $P : \{0,1\}^m \to \{0,1\}$ is a predicate that is "weakly hard to compute". Let $x^{(t)} = (x_1, ..., x_t)$ be a t-tuple of independent inputs to P. Denote $P^{(t)}(x^{(t)}) = \bigoplus_{i=1}^{t} P(x_i)$. Then the lemma asserts that P^t is "strongly hard to compute". Formally:

Theorem 7.1 (The XOR-Lemma) *Suppose that any PPTMA fails to compute $P(x)$ with probability better than $1 - \frac{1}{p(n)}$ for all but finitely many n for all auxiliary information and for a given polynomial $p(\cdot)$. Take $t(n) = p(n)^2$. Then any PPTMA fails to compute $P^{(t(n))}(x^{(t(n))})$ with probability better than $\frac{1}{2} + \frac{1}{q(n)}$ for any polynomial $q(n)$ for all auxiliary information and for all but finitely many n's (the probabilities are taken over a given distribution on the inputs $x \in \{0,1\}^m$ and the randomness of the PPTMs).*

In our application we take $x_i = \text{VIEW}^{Weak-OT}_{receiver}(c = 0, b_i)$ and $P(x_i) = b_i$. By the definition of Weak-OT, P is indeed weakly hard. The conclusion is that if $c = 0$, then the sender cannot predict the bit $b = \bigoplus_{i=1}^{p(n)^2} b_i$ from the view of $\text{OT}^{Weak-OT}$ with a non-negligible advantage, as required in a strong OT protocol.

■

8 Further Issues

The Symmetric SFE Model: In this model, both sides receive the output $f(x, y)$ at the end of the protocol. Actually, most of the works considering the questions of completeness and triviality were conducted in this setting (e.g. [10, 34, 31, 35, 33] and part of [32]). While the study of Boolean functions in this model, yielded clean and tight results [10], this is not the case when considering non-Boolean functions. In fact Kushilevitz [34] demonstrates a function that is neither complete nor trivial in the information theoretic model, so no tight characterization can be expected.

On the other hand, Kilian [31] did show that the complete functions are exactly those containing an imbedded OR. Here as well, the proof that if a function does not contain an imbedded OR then it is not complete works only for finite functions (it uses a reduction that is not polynomial time). Moreover when discussing computational security, the following example shows a function that does not contain an imbedded OR but is still complete:

Example 4 *Let $g : \{0, 1\}^n \leftarrow \{0, 1\}^n$ be a one-one one-way function, and let $x_0, x_1, y_0, y_1 \in \{0, 1\}^n$ and $c \in \{0, 1\}$. Define the function $f : \{0, 1\}^{2n+1} \times \{0, 1\}^{2n} \to \{0, 1\}^{2n}$ as follows:*

$$f((c, x_0, x_1), (y_0, y_1)) = (x_0 \oplus y_c, x_1 \oplus g(y_{1-c}))$$

The function f does not contain an imbedded Or (every row is a one-one function), but can be shown to be complete in the symmetric model. This leaves room to try and find a computational characterization in this model as well.

Probabilistic Functionalities: Kilian [32] gives combinatorial criteria also for secure evaluation of probabilistic functionalities (rather than deterministic functions). It is interesting to see if computational results can be found in this model as well. While it is known that given OT one can achieve SFE of any functionality, it is interesting what can be achieved on lesser assumptions. For example, consider a protocol in which neither party has an input at all, and in which Alice receives as output $g(z)$ for a function g and a random input z unknown to both Alice and Bob. This is a good example of an SFE of a probabilistic functionality that we call *interactive oblivious sampling* (IOS) and has useful applications. For instance, IOS can be used in order to build an oblivious transfer protocol given a secret key agreement protocol. It is not known what functions g can be obliviously sampled if the existence of OT is not guaranteed.

Acknowledgments: We are grateful to Tal Malkin for discussions regarding her work on this subject and to Ran Canetti for his helpful comments on this paper. We also thank Ronen Shaltiel for useful advice regarding hardness amplification.

References

[1] W. Aiello and J. Hastad. Statistical zero-knowledge languages can be recognized in two rounds. *Journal of Computer and Systems Sciences (JCSS)*, 42(3):327–345, 1991.

[2] A. Beimel, T. Malkin, and S. Micali. The all-or-nothing nature of two-party secure computation. In *Advances in Cryptology - CRYPTO '99, Lecture Notes in Computer Science*, volume 1666, pages 80–97. Springer, 1999.

[3] M. Bellare and S. Micali. Non-interactive oblivious transfer and applications. In *Advances in Cryptology - CRYPTO '89, Lecture Notes in Computer Science*, volume 435, pages 547–557. Springer, 1989.

[4] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *20th ACM Symposium on the Theory of Computing*, pages 1–10, 1988.

[5] C.H. Bennett, G. Brassard, C. Crepeau, and M.H. Skubiszewska. Practical quantum oblivious transfer protocols. In *Advances in Cryptology - CRYPTO '91, Lecture Notes in Computer Science*, volume 576, pages 351–366. Springer, 1992.

[6] R. Boppana, J. Hastad, and S. Zachos. Does co-np have short interactive proofs? *Information Processing Letters*, 25(2):127–132, 1987.

[7] C. Cachin, C. Crepeau, and J. Marcil. Oblivious transfer with a memory-bound receiver. In *39th IEEE Symposium on Foundations of Computer Science*, pages 493–502, 1995.

[8] R. Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000.

[9] D. Chaum, C. Crepeau, and I. Damgard. Multiparty unconditionally secure protocols. In *20th ACM Symposium on the Theory of Computing*, pages 11–19, 1988.

[10] B. Chor and E. Kushilevitz. A zero-one law for boolean privacy. *SIAM Journal on Disc. Math.*, 4(1):36–47, 1991. preliminary version in STOC 89.

[11] C. Crepeau. Equivalence between two flavours of oblivious transfers. In *Advances in Cryptology - CRYPTO '87, Lecture Notes in Computer Science*, volume 293, pages 350–354. Springer-Verlag, 1987.

[12] C. Crepeau and J. Kilian. Achieving oblivious transfer using weakened security assumptions. In *29th IEEE Symposium on Foundations of Computer Science*, pages 42–52, 1988.

[13] I. Damgard, 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, Lecture Notes in Computer Science*, volume 1592, pages 56–73, 1999.

[14] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.

[15] M. Fitzi, J.A. Garay, U.M. Maurer, and R. Ostrovsky:. Minimal complete primitives for secure multi-party computation. In *Advances in Cryptology - CRYPTO '01, Lecture Notes in Computer Science*, volume 2139, pages 80–100. Springer, 2001.

[16] L. Fortnow. The complexity of perfect zero-knowledge. *Advances in Computing Research (editor S. Micali)*, 18, 1989.

[17] Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The relationship between public key encryption and oblivious transfer. In *41st IEEE Symposium on Foundations of Computer Science*, pages 325–335, 2000.

[18] O. Goldreich. *Foundations of Cryptography.* Cambridge University Press, 2001.

[19] O. Goldreich. Foundations of cryptography - volume 2. Working Draft, available at www.wisdom.weizmann.ac.il/oded/foc-vol2.html, 2002.

[20] O. Goldreich and L.A. Levin. A hard-core predicate for all one-way functions. In *21st ACM Symposium on the Theory of Computing*, pages 25–32, 1989.

[21] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game - a completeness theorem for protocols with honest majority. In *19th ACM Symposium on the Theory of Computing*, pages 218–229, 1987.

[22] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity, or all languages in np have zero-knowledge proof systems. *Journal of the ACM*, 38:691–729, 1 1991.

[23] O. Goldreich, N. Nisan, and A. Wigderson. On Yao's XOR-lemma. In *Electronic Colloquium on Computational Complexity (ECCC) (50)*, volume 2, 1995.

[24] O. Goldreich and R. Vainish. How to solve any protocol problem - an efficiency improvement. In *Advances in Cryptology - CRYPTO '87, Lecture Notes in Computer Science*, volume 293, pages 73–86. Springer-Verlag, 1987.

[25] M. Hirt and U. Maurer. Player simulation and general adversary structures in perfect multiparty computation. *Journal of Cryptology*, 13(1):31–60, 2000.

[26] R. Impagliazzo. Hard-core distributions for somewhat hard problems. In *36th IEEE Symposium on Foundations of Computer Science*, pages 538–545, 1995.

[27] R. Impagliazzo. A personal view of average-case complexity. In *10th Annual Structure in Complexity Theory Conference*, pages 134–147. IEEE Computer Society Press, 1995.

[28] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. In *30th IEEE Symposium on Foundations of Computer Science*, pages 230–235, 1989.

[29] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *21st ACM Symposium on the Theory of Computing*, pages 44–61, 1989.

[30] J. Kilian. Founding cryptography on oblivious transfer. In *20th ACM Symposium on the Theory of Computing*, pages 20–31, 1988.

[31] J. Kilian. A general completeness theorem for two-party games. In *23rd ACM Symposium on the Theory of Computing*, pages 553–560, 1991.

[32] J. Kilian. More general completeness theorems for secure two-party computation. In *32nd ACM Symposium on the Theory of Computing*, pages 316–324, 2000.

[33] J. Kilian, E. Kushilevitz, S. Micali, and R. Ostrovsky. Reducibility and completeness in private computations. *SIAM Journal of Computing*, 28(4):1189–1208, 2000.

[34] E. Kushilevitz. Privacy and communication complexity. *SIAM Journal on Disc. Math.*, 5(2):273–284, 1992. preliminary version in FOCS 89.

[35] E. Kushilevitz, S. Micali, and R. Ostrovsky. Reducibility and completeness in multi-party private computations. In *35th IEEE Symposium on Foundations of Computer Science*, pages 478–489, 1994.

[36] L. A. Levin. One-way functions and pseudorandom generators. *Combinatorica*, 7:357–363, 1987.

[37] U. Maurer. Information-theoretic cryptography. In *Advances in Cryptology — CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 47–64. Springer-Verlag, 1999.

[38] S. Micali and P. Rogaway. Secure computation. In *Advances in Cryptology - CRYPTO '91, Lecture Notes in Computer Science*, volume 576, pages 392–404. Springer, 1991.

[39] M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *SIAM Symposium on Discrete Algorithms (SODA 2001)*, pages 448–457, 2001.

[40] T. Okamoto. On relationships between statistical zero-knowledge proofs. *Journal of Computer and Systems Sciences (JCSS)*, 60(1):47–108, 2000.

[41] R. Ostrovsky and A. Wigderson. One-way fuctions are essential for non-trivial zero-knowledge. In *Second Israel Symposium on Theory of Computing Systems, ISTCS 93, Proceedings. IEEE Computer Society*, pages 3–17, 1993.

[42] M. O. Rabin. How to exchange secrets by oblivious transfer. TR-81, Harvard, 1981.

[43] A. C. Yao. Protocols for secure computations. In *23rd IEEE Symposium on Foundations of Computer Science*, pages 160–164, 1982.

[44] A. C. Yao. Theory and application of trapdoor functions. In *23rd IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982.

[45] A. C. Yao. How to generate and exchange secrets. In *27th IEEE Symposium on Foundations of Computer Science*, pages 162–167, 1986.

A Separating UA-Eff-$\mathcal{SFE}$ and IT-Eff-$\mathcal{SFE}$

In Example 1 we show a function that is in IT-Eff-$\mathcal{SFE}$ but on the other hand in Co-$\mathcal{SFE}$-C (under a computational assumption). This shows a separation between the computational model and the unbounded adversaries and information theoretic models.

An interesting question is whether a similar separation can be found between the unbounded adversaries and the fully information theoretic model. That is, is there a function with no insecure minor that is not in UA-Eff-$\mathcal{SFE}$ under reasonable assumptions?[26] We give a partial answer to this question, starting by giving an example of such a function in the malicious adversaries model.

Example 5 *Let Φ be a 3-CNF formula and A be an assignment to its variables. Let $\Phi(A)$ denote the value of the formula Φ under the assignment A. Consider the function:*

$$f_{3-SAT}(\cdot, (\Phi, A)) = (\Phi, \Phi(A))$$

(Alice gets no input while Bob gets input $y = (\Phi, A)$.[27]

[26] Recall that functions with no insecure minor are in IT-Eff-$\mathcal{SFE}$.

[27] Note that the choice of 3-SAT is immaterial and any $\mathcal{NP}$-complete language would have sufficed

In the semi-honest model f_{3-SAT} has a trivial SFE protocol where Bob simply sends the output $(\Phi, \Phi(A))$ to Alice. In the malicious model however, it is not clear what can be done to prevent Bob from cheating. Furthermore we claim the following:

Claim A.1 *If there exists a malicious SFE protocol for f_{3-SAT} in the unbounded adversaries model then the polynomial hierarchy collapses to the second level.*

The Claim is proved by showing that a malicious SFE protocol for f_{3-SAT} in the unbounded adversaries is actually a perfect zero knowledge proof for the following $\mathcal{NP}$-complete language: 3-SAT $= \{\Phi | \Phi$ has a satisfying assignment $\}$. The correctness of the SFE protocol assures the verifier (playing Alice) that upon receiving output $(\Phi, 1)$, there is indeed a satisfying assignment for Φ. On the other hand, the security of the SFE protocol assures that the view of Alice can be perfectly simulated by an efficient simulator. Since 3-SAT is also $\mathcal{NP}$-complete we deduce that all $\mathcal{NP}$ languages have perfect zero knowledge proofs. It is known that every language with a perfect zero knowledge proof is also in co-$\mathcal{AM}$ [16, 1, 40], thus $\mathcal{NP} \subseteq$ co-$\mathcal{AM}$. Combining this claim with a result of Boppana et al. [6] we get that f_{3-SAT} does not have an SFE protocol in the malicious unbounded adversaries model unless the polynomial time hierarchy collapses to the second level. Thus UA-Eff-$\mathcal{SFE} \subsetneq$ IT-Eff-$\mathcal{SFE}$ in the malicious case under reasonable assumptions.

We further ask whether such a separation exists in the semi-honest case as well. The following example can be viewed as some indication that it is unlikely that IT-Eff-$\mathcal{SFE}$ = UA-Eff-$\mathcal{SFE}$ unconditionally.

Example 6 *Let $g_0, g_1 : \{0,1\}^n \to \{0,1\}^n$ (for every n) be two one-to-one functions. Define:*

$$L_{g_0,g_1} = \{(z_0, z_1) | \text{ there exists } y \text{ such that } z_0 = g_0(y) \text{ and } z_1 = g_1(y)\}$$

Also define:

$$f_{g_0,g_1}(x,y) = \begin{cases} g_0(y) & x = 0 \\ g_1(y) & x = 1 \end{cases}$$

Clearly $f_{g_0,g_1}(x,y) \in$ IT-Eff-$\mathcal{SFE}$ since an all powerful Alice learns y from either $g_i(y)$ and therefore Bob may simply send y to Alice. On the other hand we claim the following:

Claim A.2 *If there exist g_0, g_1 such that $L_{g_0,g_1} \notin \mathcal{PZK}$ (L_{g_0,g_1} does* not *have a perfect zero knowledge proof) then $f_{g_0,g_1}(x,y) \notin$ UA-Eff-$\mathcal{SFE}$.*

Proof. Suppose that $f_{g_0,g_1}(x,y) \in$ UA-Eff-$\mathcal{SFE}$, then we will show a perfect zero knowledge protocol for L_{g_0,g_1}. The assumption states that there is a protocol $\Pi(x,y)$ that always outputs $f_{g_0,g_1}(x,y)$, and there are simulators S_A and S_B that perfectly simulate the respective views of Π. Perfect simulation means that, in particular, every possible real view v of $\Pi(x,y)$ is also a possible output of the simulator (of both simulators), and every possible output of the simulator is also a possible real view. Notice that the simulator $S_B(y)$ outputs a view of the conversation between Alice and Bob that is independent of the value x. Hence, for any such simulated view v there is an identical real conversation of $\Pi(x=1,y)$ and also a real conversation of $\Pi(x=0,y)$. Therefore, the view v is also a possible output of $S_A(x, g_x(y))$ for $x=0$ and for $x=1$.

The perfect zero knowledge proof for L_{g_0,g_1} goes as follows: Given $(z_0,z_1) \in L_{g_0,g_1}$, with an input y corresponding to z_0 and z_1, the prover generates a random view v of a conversation of the protocol $\Pi(x,y)$ for any x ($x=0$ will do). The prover sends the view v to the verifier. The verifier then chooses a random bit x and challenges the prover to reveal how the view sent may correspond to an output of $S_A(x, z_x)$. A non cheating prover can always provide the random coins that give $S_A(x, z_x) = v$, but if $(z_0, z_1) \notin L_{g_0,g_1}$ then by the correctness of the SFE protocol Π, the prover will fail on at least one of the possible challenges, and will be caught with probability at least $\frac{1}{2}$. The above protocol is also zero knowledge since the verifier's view may be perfectly simulated, simply by choosing a random x and setting the view to be $v = S_A(x, g_x(y))$. ■

The above claim is not in itself a separation of UA-Eff-$\mathcal{SFE}$ from IT-Eff-$\mathcal{SFE}$. It merely points out that if there is no separation, then all languages of the type L_{g_0,g_1} have perfect zero knowledge proofs. We view this as an interesting consequence as we are unaware of such a generic construction.