

Completeness in Two-Party Secure Computation: A Computational View*

Danny Harnik, Moni Naor, and Omer Reingold
Department of Computer Science and Applied Mathematics,
Weizmann Institute, Rehovot, 76100 Israel
{danny.harnik,moni.naor,omer.reingold}@weizmann.ac.il

Alon Rosen
DEAS, Harvard University,
Cambridge, MA 02138, U.S.A.
alon@eecs.harvard.edu

Communicated by Ran Canetti

Received 7 September 2003 and revised 30 November 2005
Online publication 24 May 2006

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*

* A preliminary version of this work appeared in *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC 04)*. Danny Harnik’s research was supported in part by a grant from the Israel Science Foundation. Moni Naor is an Incumbent of the Walter and Elise Haas Career Development Chair, whose research was supported in part by a grant from the Israel Science Foundation. Omer Reingold is an Incumbent of the Walter and Elise Haas Career Development Chair, most of whose research was performed while at AT&T Labs and the Institute for Advanced Study, Princeton, NJ. Most of this by research by Alon Rosen was performed while at the Weizmann Institute of Science, Israel.

(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.

Key words. Oblivious transfer, One-way function, Secure function evaluation.

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, as a first step, 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. The results are then examined in the real world where malicious behavior is allowed.²

1.1. Completeness in SFE

A major component in the construction of SFE protocols is the Oblivious Transfer (OT) protocol, Rabin’s brainchild [44]. OT refers to several different versions of SFE protocols, all of which turned out to be equivalent. For instance, consider the 1-2-OT [15], 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 $f_{\text{OT}}(c, (b_0, b_1)) = b_c$.

OT plays a key role in secure computation since it was shown to be *complete for SFE* [45], [47], [22], [25], [32] (see Section 2.3 for details), 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 and, indeed, several implementations of OT protocols have been suggested relying on various computational assumptions.⁴

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 (see Section 1.5).

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

⁴ For example, OT was based on general assumptions (e.g. [15], [20], and [18]) or on specific assumptions (e.g. the Diffie–Hellman assumptions [3], [41])

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. We denote by $\mathcal{SFE}\text{-C}$ the set of functions which are complete for SFE. This set in particular contains the function f_{OT} . We also denote by $\text{Eff-}\mathcal{SFE}$ the set of functions for which there exists (efficient) SFE. The set $\text{Eff-}\mathcal{SFE}$ 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 $\mathcal{SFE}\text{-C}$? The ability of identifying functions in $\mathcal{SFE}\text{-C}$ can be particularly useful as a tool for implementing OT: If we are able to design SFE for a function f which is $\mathcal{SFE}\text{-C}$ we are immediately guaranteed an implementation of OT (we show a specific example in Section 6.1).
2. How do the two sets relate? One possibility which is consistent with our current knowledge is that $\mathcal{SFE}\text{-C} = \text{Eff-}\mathcal{SFE}$ 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 $\mathcal{SFE}\text{-C} \cap \text{Eff-}\mathcal{SFE} = \emptyset$. 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 $f \in \text{Eff-}\mathcal{SFE}$ for functions $f \notin \mathcal{SFE}\text{-C}$. Are any of these assumptions useful “fall backs” in the unfortunate scenario where OT does not exist?

1.2. Related Work

The above questions were investigated in a large body of work [4], [11], [36], [33], [2], [37], [35], [34], [16].⁵ This study was mainly conducted from an information theoretic point of view. Specifically, most of these papers consider computationally unbounded parties. Matching the definition of SFE, the notion of completeness is usually information theoretic as well. From this perspective this line of research obtains very strong results. Loosely, these papers classify functions as either complete or in $\text{Eff-}\mathcal{SFE}$ unconditionally⁶ and provide combinatorial properties for separating the two. Formalizing the various models considered by this line of work (and specifically the various notions of SFE and completeness) is beyond the scope of this paper.⁷ Instead, we concentrate here on the most relevant combinatorial criteria provided by these works:

Imbedded OR. The papers initiating this area of research are those of Chor and Kushilevitz [11], [36] and Kilian [33]. They consider the symmetric version of SFE and give criteria for having unconditional SFE and for completeness. In particular, Kilian [33] proves 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 $f(x_0, y_0) = f(x_0, y_1) =$

⁵ It should be noted that many of these works do not mention completeness explicitly.

⁶ By unconditionally we mean that no hardness assumption is needed.

⁷ Section 6.3 contains a discussion on different models of security.

Table 1. Imbedded OR and insecure minor.

	Imbedded OR		Insecure minor	
	y_0	y_1	y_0	y_1
x_0	a	a	a	a
x_1	a	b	b	c

$f(x_1, y_0) \neq f(x_1, y_1)$ (see Table 1).⁸ When discussing Boolean functions, it turns out that the non-complete functions (functions not containing an imbedded OR) are in Eff-SFE unconditionally [11].

Insecure Minor. Beimel et al. [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 $f(x_1, y_0) \neq f(x_1, y_1)$ (see Table 1),⁹ and showed that every function with an insecure minor is complete, while every other function has a trivial SFE. This work was conducted under computational definitions of security, using the compiler of Goldreich et al. [23] to assure security against malicious parties rather than semi-honest parties (as we do in this work, see Section 1.5). In order to assure polynomial running time of all parties, this work was restricted to functions over a domain of small (or constant) size. This provides a very complete answer to the questions posed above under this restriction.

Other Work. Other works include the generalization of the criteria to the case of multi-parties (in the symmetric case) [37], [35] and probabilistic functionalities [34] (as opposed to deterministic functions). Completeness in multi-party computation was also studied with regard to a measure of cardinality (the number of participating parties) [16]. Another work that discusses a computational property rather than a combinatorial one is [9], where a criterion for universally composable secure computation is given.

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 computationally unbounded parties or functions that are only defined on a constant domain size), 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!

⁸ The name “imbedded OR” comes from the fact that the function projected to these four inputs looks like an OR or an AND gate.

⁹ Note that every imbedded OR is also an insecure minor but not vice versa.

The problem can be illustrated nicely by looking at a special case of functions with no insecure minor—the one-to-one functions.¹⁰ By the insecure minor criterion a one-to-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 priori, however, the running time of this simulator is exponential in n . This is acceptable if we think of n as small (as Beimel et al. [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 in [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.¹¹

Example 1. Let g be a one-to-one one-way function ($g: \{0, 1\}^n \rightarrow \{0, 1\}^n$ for all n). The function $f: \{0, 1\}^1 \times \{0, 1\}^{2n} \rightarrow \{0, 1\}^{2n+1}$ 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 cannot 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 there exist simple criteria at all that can capture the above notions in the general computational scenario? We answer this question positively.

This paper presents computational criteria for being in $SFE-C$ and $Eff-SFE$. These criteria are computational in nature and hold for functions of unbounded input length. Also, they are very close to being complementary to each other, thus almost fully categorizing all efficiently computable functions.

¹⁰ In one-to-one functions $f(x_0, y_0) \neq f(x_0, y_1)$ for all x_0, y_0, y_1 . So no insecure minor can exist.

¹¹ 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 [30].

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 (some-what) 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 a value at one row is equivalent learning it at any other row.

Superficially, the two properties may seem exactly complementary to each other. However the exact formulations (Definitions 4.1 and 3.1) leave 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 ($f \in \text{Eff-SFE}$).
2. If f is row non-transitive then it is complete ($f \in \text{SFE-C}$).

Outline of Proof. In the following discussion we try to present the main ideas for the proofs. 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).¹² 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 [21]. 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 hardcore bit with better than negligible probability. Bob learns essentially nothing in the process.

The actual proof of security calls for a more careful treatment: To begin with, we note that our use of the GL hardcore bit is somewhat more involved as it is not applied to a one-way function.¹³ Furthermore, the proof that the GL bit is hard to predict requires

¹² In the semi-honest model, Naive-OT is equivalent to the so-called Noisy-OT, that in turn was shown to be equivalent to 1-2-OT [12].

¹³ 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.

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 $\mathcal{SFE}\text{-C}$ and $\text{Eff-}\mathcal{SFE}$). 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-OT¹⁴ 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 [46] (see [24]).

Note that an implementation of OT was shown to guarantee secure computation in further computational models such as multi-party secure computation [22] (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. The Results in the Malicious Model

Our paper and results concentrated thus far on 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 from the compiler of Goldreich, Micali and Wigderson (GMW) [23], 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 also carry over to the real world where 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 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 an SFE protocol must refer only to sending messages and tossing coins, without the 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 in [2] and [34].¹⁵

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 of an SFE for a complete function implies semi-honest OT that in turn

¹⁴ The notion of Weak-OT in this paper (defined in Section 3) is different from various versions of OT with the same name defined in other papers (e.g. [14] and [34]).

¹⁵ For example the OR of two bits is complete in the semi-honest case, but not so in the malicious model.

implies the existence of one-way functions (e.g. [29]). So the completeness part of the theorem holds unconditionally:

Theorem 1.2. *If a row non-transitive function has an efficient SFE protocol in the semi-honest model, then all polynomial time functions have SFE protocols that are 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 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.

1.6. Paper Organization

In Section 2 we present some relevant definitions and notations, including the notions of reductions and completeness. In Sections 3 and 4 we present our criteria and main theorems. Section 5 elaborates on the applicability of our results in the malicious model. Section 6 contains a discussion on the meaning of the results in this paper as well as a general discussion regarding the computational model versus information theoretic ones. Section 7 mentions some further issues and questions.

2. Formal Setting

Some general notations for this paper: PPTM stands for Probabilistic Polynomial Time Turing Machine. By a **Distribution 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.

In this paper we choose to define security to be against non-uniform adversaries. These are presented as Turing machines that also receive 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 $w \in \{0, 1\}^*$.

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 6.2 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 $\{X_s\} \stackrel{c}{\approx} \{Y_s\}$) if for every PPTMA M , 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, Y_s and the randomness of M .

2.1. SFE

Discussing the various definitions of SFE is beyond the scope of this paper. The reader is referred to [8] and [20] 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 [20]. 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 security parameter for the SFE protocol (usually denoted n).
- The input length (combined length of the inputs) is bounded by a polynomial of the security parameter.
- For sake of simplicity and without loss of generality, assume that for input length $l(n)$ the output is always of length $m(n)$ (where $m(\cdot)$ is a polynomial). This assumption is justified by a padding argument.¹⁶
- Assume that all parties know n and the length of the inputs. Also, when giving partial inputs to various PPTMs, 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 (in the Semi-Honest Model)). Let $f: \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}^*$ 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 $x, y \in \{0, 1\}^*$, $\Pi(x, y) = f(x, y)$.

¹⁶ 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.

2. Security:

- Bob’s Privacy: There exists a PPTM S_A such that

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

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

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

Denote the set of functions that have an efficient SFE protocol by Eff-SFE .

SFE in the Malicious Model. We stress that the above definition (adapted from Definition 7.2.1 of [20]) applies only for the semi-honest model. For the definition in the case of malicious parties we refer the reader to [8] and [20]. In general, this definition introduces an “ideal box” that evaluates f , where both parties enter their respective inputs and Alice receives the output. For a protocol to be an SFE the parties should gain nothing in the actual real computation 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 “ideal-real” approach can capture both the malicious and semi-honest models.

However, we want to advocate a definition that is slightly more stringent than that of [8] and [20]. As pointed out by Ishai [31], one would like SFE in the malicious world to be harder to achieve than SFE in the semi-honest world. Nonetheless this is not the case for the definitions given in [8] and [20], and in fact the two models turn out to be incomparable (since the simulator is given strictly more power in the malicious model). As an example, the OR function is complete in the semi-honest model but has a very simple unconditional SFE in the malicious model (the parties can simply always use the input 0 and thus always learn the other sides input). We therefore prefer to say that a protocol is a **stringent-SFE** in the malicious model if it follows the ideal-real definition of [8] and [20] with the additional constraint that it is also an SFE in the semi-honest model as defined above.

Denote the set of functions that have an SFE protocol by Eff-SFE .

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 evaluate any function securely. 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.¹⁷
- 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

¹⁷ Bob does not see Alice’s interaction with the ideal box and vice versa.

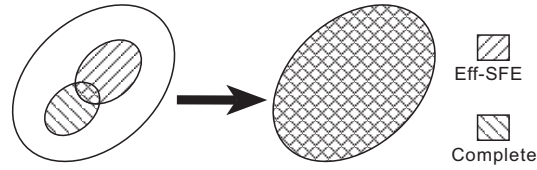


Fig. 1. The two families $\mathcal{SFE-C}$ and Eff-SFE may have no intersection, but if they intersect (as is shown in the figure) then all efficiently computable functions have SFE protocols and also $\mathcal{SFE-C} = \text{Eff-SFE}$.

some function f . This means that the two parties can compute local inputs for f (one acting as Alice and the other as Bob¹⁸) 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 the definition 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_g^f (polynomial in the length of the input to g) with access to an ideal box for evaluating f such that the protocol Π_g^f 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: $\mathcal{SFE-C} = \{f \mid f \text{ is SFE-Complete}\}$.

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.

The sets $\mathcal{SFE-C}$ and Eff-SFE have the following two possibilities: either the two sets are distinct (have no intersection) or the two sets have an intersection. However, the latter case implies that all efficiently computable functions have an SFE protocol, and thus $\mathcal{SFE-C} = \text{Eff-SFE}$ and they contain all efficiently computable functions (see Fig. 1). This is indeed the case, for instance, if efficient protocols for OT (defined in the next section) exist. Altogether, either $\mathcal{SFE-C}$ and Eff-SFE do not intersect or $\mathcal{SFE-C} = \text{Eff-SFE}$ 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 from 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 seems reasonable (or at least not surprising) that

¹⁸ The parties may switch their roles in various calls to the ideal box.

there is an intersection and that all functions are both SFE-complete and in Eff-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 \mathcal{L} ($\mathcal{NL-C} \cap \mathcal{L} = \emptyset$).

2.3. OT

A central component in the construction of SFE protocols is the OT protocol. OT refers to several equivalent versions of two-party protocols. For example, an important formulation is the 1-2-OT (due to Even et al. [15]), 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 [44]), and was shown to be equivalent to 1-2-OT in [12]. 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 does not 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 and 1-2-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 property was shown in a series of works for different models: The ideas are attributed to Yao [45], [47] originally for the semi-honest model, and through the GMW compiler [23] (originally published in 1987) it is possible to get a protocol for the malicious model. Goldreich et al. [22] then extended the work to the multi-party case. Goldreich and Vainish [25] showed a secure computation protocol based solely on OT calls,¹⁹ working only in the semi-honest model, and Kilian [32] presented a construction based only on OT in the malicious model.

In turn, OT can be constructed from trapdoor permutations and public key encryption with particular "niceness" properties [15], [20], [18] and various specific intractability assumptions (e.g. the Diffie–Hellman assumptions [3], [41]). 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 [30], [28], [18].

¹⁹ Previous results were also based on other computational assumptions (that are equivalent to the existence of one-way functions).

3. Criterion for Completeness

Our main result presents criteria for functions to be complete or in Eff-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.

Definition 3.1. 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 .²⁰

We note that the notion of hardness in row non-transitivity is very different from that of a one-way function. To begin with, 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). Furthermore, 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).

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

To prove that every row non-transitive function is complete 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,

²⁰ Recall that a PPTMA is a PPTM with auxiliary information, that is polynomial time in the security parameter.

the following hold:

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

$$\Pr[A(\text{VIEW}_{\text{receiver}}^{\text{weak-OT}}(0, b), w) = b] < 1 - \frac{1}{p'(n)}.$$

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

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

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 3.1. 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 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 $1 - 1/p(n)$ for some polynomial $p(\cdot)$. This is equivalent to the weak version of Goldreich–Levin's (GL) 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 .

To conclude the proof of Theorem 3.1 suppose that there exists a PPTMA A that predicts the bit b from the receiver's view, with probability better than $1 - 1/q(n)$ 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 $\langle f(x_1, y), r \rangle$ given only $x_0, x_1, f(x_0, y)$ with probability better than $1 - 1/q(n)$ for all polynomials $q(\cdot)$, and this contradicts Lemma 3.3. \square

Proof of Lemma 3.3. 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,

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

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

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

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 (given 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 $\Pr_{y \in D_y(1^n)}[E_n] > 1 - 1/2p(n)$ 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 i th 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 1 in the i th place and 0 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 $1 - 2 \cdot (1 - \frac{2}{10}) = \frac{8}{10}$. 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 $\Pr[E_n] \cdot (1 - 2^{-\Omega(n)}) > (1 - 1/2p(n))(1 - 2^{-\Omega(n)}) > 1 - 1/p(n)$, contradicting the non-transitivity of f . \square

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 3.1). An alternative approach would be first to 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 GL 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 $1 - 1/p(n)$ 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 Insecure Minor 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 $f(x_1, y_0) \neq f(x_1, y_1)$.²¹ 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),

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

and hence the function f is also computational row non-transitive.

3.1. 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 [13] and [14] that were all information theoretic. While the construction used here is essentially the same, it works against computational adversaries and thus requires a more complex proof using 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 $1 - 1/p(n)$ 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 $t(n) = p(n)^2$.

²¹ We note that this definition is for functions of constant input length. The generalization to functions on unbounded input length requires having 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.

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

1. The Sender randomly chooses $t(n)$ random bits $b_1, \dots, b_{t(n)-1} \in_R \{0, 1\}$ 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 $1/q(n)$ 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 $1/t(n)q(n)$ 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 [46]), with formal proofs presented in [38], [27], and [24]. The XOR lemma states the following:

Suppose that $P: \{0, 1\}^m \rightarrow \{0, 1\}$ is a predicate that is "weakly hard to compute". Let $x^{(t)} = (x_1, \dots, x_t)$ 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 3.5 (The XOR Lemma). *Suppose that any PPTMA fails to compute $P(x)$ with probability better than $1 - 1/p(n)$ 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 $\frac{1}{2} + 1/q(n)$ 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 general, for our application we take $x_i = \text{VIEW}_{\text{receiver}}^{\text{Weak-OT}}(c = 0, b_i)$ and define $P(x_i) = b_i$. 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 OT^{Weak-OT} with a non-negligible advantage, as required in a strong OT protocol. However, note that the view x_i does not necessarily define the bit b_i , so the predicate P is not well defined in this case. Instead we slightly restate the XOR lemma to be of the following general form: given a predicate $P: \{0, 1\}^l \rightarrow \{0, 1\}$ and an efficiently computable function $g: \{0, 1\}^l \rightarrow \{0, 1\}^m$, if it is weakly hard to compute $P(y_i)$ given only $x_i = g(y_i)$ then it is "strongly hard to compute" $P^{(t)}(y^{(t)})$ given only $x^{(t)}$. \square

4. Criterion for Eff-SFE

We now present our criterion for a function to be unconditionally in Eff-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. 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 emphasize 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-SFE.²²*

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

$\Pi_f(x, y)$

1. Bob chooses a random $\hat{x} \in \{0, 1\}^n$ 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 procedure is easily shown to be an SFE protocol for f . The correctness follows since Alice learns the value $M(\hat{x}, x, f(\hat{x}, y)) = f(x, y)$ 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 $M(x, \hat{x}, f(x, y)) = f(\hat{x}, y)$. \square

5. The Semi-Honest versus the Malicious Model

Throughout the paper we allow ourselves to assume that Alice and Bob are semi-honest. 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 GMW compiler [23], which

²² 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).

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):

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 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. Also, note that unlike in [2], the GMW compiler is needed for both items of Theorem 5.1 (see discussion below).

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 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 [29], the existence of semi-honest OT implies the existence of one-way functions.²³ 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 [45], 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 . \square

Using the Stringent Definition of SFE. When using the *stringent-SFE* definition (see Section 2.1), the above theorems take a slightly different form. Recall that the stringent definition requires that in order for an SFE protocol to be secure (in the malicious model) it must also be a semi-honest SFE. The statement in the malicious model then becomes more symmetric:

Theorem 5.3. *If a row non-transitive function has an efficient (stringent-)SFE protocol, then all polynomial time functions have (stringent-)SFE protocols.*

²³ Actually, in [29] it was only shown that several other primitives (not OT) imply one-way functions, for example, this was shown for bit commitment. However, in [32] among other things it shows that OT implies bit commitment and hence also implies one-way functions.

Row Transitive Functions in the Malicious Model. Unlike the theorem for completeness, 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, the following claim shows that under certain (plausible) assumptions, 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 $\{f_i: D_i \rightarrow \{0, 1\}^*\}_{i \in \bar{I}}$ has **one-way instances** if:

1. Easy to sample and compute: For each of the following tasks there exists a PPTM: sampling $i \in \bar{I}$, sampling $x \in D_i$ and computing $f_i(x)$.
2. Some functions are hard to invert: For every 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 the probability is taken over $x \in D_i$ and the randomness of A .

This weak notion of one-wayness was defined in [19] as part of the presentation of [43]. We can now state the following claim:

Claim 5.4. *Suppose that there exist no collections of functions with one-way instances and also $\mathcal{NP} \not\subseteq \mathcal{BPP}$, then there exist functions f that have semi-honest SFE but no malicious SFE.*

Proof. A theorem of Ostrovsky and Wigderson [43] 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 containing the union of the edges in an HC C and a basic edge set E (G returns a graph that in particular contains an HC). Define the function $f(x, y) = G(y)$, where $y = (C, E)$ (let $f(y) = \perp$ if C is not an HC). The function f is definitely a row transitive function as it disregards the input x 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 zero knowledge 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 an HC 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 an HC. However, since an HC is an NP-complete problem, and we assumed that $\mathcal{NP} \not\subseteq \mathcal{BPP}$, then an HC 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. \square

Semi-Honest versus 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 [34] showed that the criterion for completeness in the malicious model is different from 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. Discussions

6.1. 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 $\text{Eff-}\mathcal{SFE}$. 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 Fig. 2). 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 6.2 contains a discussion on this gap). Moreover, it seems that this gap cannot be closed

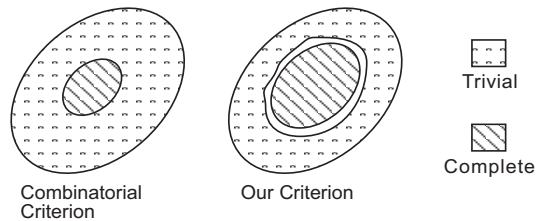


Fig. 2. The area inside the outer circle depicts all efficiently computable functions. The left picture shows the tight characterization of functions according to the insecure minor criterion, while the right picture shows the characterization according to the “row transitivity” criteria.

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 [28] that describes five possible 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 ($\mathcal{SFE}\text{-C} = \text{Eff-}\mathcal{SFE}$). 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 $f \in \mathcal{SFE}\text{-C}$. Consider the following example:

Example 2. Consider $f_p(g, y) = g^y \pmod p$, where p is a prime and q is a prime dividing $p - 1$.²⁴ 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 $g^y \pmod p$. 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 r th 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–Hellman (CDH) assumption, f is computational 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 1-2-OT: Let Bob be the sender holding secret bits b_0, b_1 and let 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 sends $z, h(g_0^y) \oplus b_0, h(g_1^y) \oplus b_1$ to Alice ($h(\cdot)$ again stands for the GL hardcore bit). Finally,

²⁴ For convenience choose p to be of the form $p = 2q + 1$ for prime q .

Alice computes $z^{r^{-1}} = g_c^y$. Alice can compute $h(g_c^y)$ and subsequently learn the secret bit b_c . However, she practically learns nothing about b_{1-c} .

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 simply to prove that a function is SFE-complete.

Example 3. Consider the function $f_N(x, y) = (x + y)^3 \bmod N$, 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 $q - 1$.²⁵

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 (for $e = 3$) f_N is row non-transitive. The general idea is that if f_N is row transitive, then given $a = (z - 1)^3 \bmod N$ one can compute the value $z - 1$, thus contradicting the hardness of RSA. This is done by computing $b = (z + 1)^3 \bmod N$ from the value a (this is possible given the row transitivity). Then the value z can be found, since $a - b = 2z^2 + 2 \bmod N$, giving the value of $z^2 \bmod N$ and in turn $z = z^3/z^2 \bmod N$. 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.

6.2. The Gap between the Criteria

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

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 $\text{Eff-}\mathcal{SFE}$. 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 possess some “unnatural” behavior 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 value $f(x_1, y)$ 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

²⁵ Assume here that N is given by an external source.

uniform adversary (PPTM): It is possible to define SFE to be secure only against uniform adversaries when working in the semi-honest model.²⁶ 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 value $f(x_1, y)$ 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-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 $1 - 1/q(n)$, 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.

6.3. Information Theoretic versus Computational—A Discussion

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.²⁷ 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 whether what it receives is the real or simulated view.²⁸ The difference between the three models is in the computational power of each of the entities.

²⁶ This is since composition theorems hold for uniform machines only under the semi-honest model.

²⁷ Note the three models as well as their names, represent the authors personal view of the main possible models.

²⁸ We note that the roles of the parties, simulators and distinguishers as described above corresponds to the semi-honest model. However, these entities may be generalized to apply to the malicious model as well.

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 OT ([4], [11] 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 compute the function f efficiently 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 (of comparable efficiency to the parties). 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 do not gain any information at all from participating in the protocol. Indeed, this model is advocated throughout the literature (e.g. [8], [20], [26] and [40]).

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 $\text{Eff-}\mathcal{SFE}$ 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 evaluate all functions securely. 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. [13]), quantum channels (e.g. [5]), the bounded storage model (e.g. [7]) and multi-party computations (e.g. [4] and [10]); see other examples [39].

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).

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 [23] 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 set-up 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 $IT\text{-}\mathcal{SFE}\text{-}C$ and $IT\text{-}Eff\text{-}\mathcal{SFE}$ the sets of complete functions and those with efficient SFE protocols in the fully information theoretic model, denote by $UA\text{-}\mathcal{SFE}\text{-}C$ and $UA\text{-}Eff\text{-}\mathcal{SFE}$ those sets in the unbounded adversaries model and denote by $Co\text{-}\mathcal{SFE}\text{-}C$ and $Co\text{-}Eff\text{-}\mathcal{SFE}$ those 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 the fully information theoretic model and in the computational model. Therefore we have that $UA\text{-}Eff\text{-}\mathcal{SFE} \subseteq IT\text{-}Eff\text{-}\mathcal{SFE}$ and also $UA\text{-}Eff\text{-}\mathcal{SFE} \subseteq Co\text{-}Eff\text{-}\mathcal{SFE}$.
- In the work of Beimel et al. [2] it is shown that $IT\text{-}\mathcal{SFE}\text{-}C \cap IT\text{-}Eff\text{-}\mathcal{SFE} = \emptyset$ and also that all functions are in either $IT\text{-}\mathcal{SFE}\text{-}C$ or $IT\text{-}Eff\text{-}\mathcal{SFE}$ (depending if they have an insecure minor or not²⁹).

²⁹ An 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.

- Since the reduction from OT to a function with an insecure minor is efficient, we have that $\text{IT-SFE-C} = \text{UA-SFE-C}$ and they include exactly all functions with an insecure minor.³⁰
- In Example 1 we showed a function that is in IT-Eff-SFE but on the other hand also in Co-SFE-C (under a computational assumption). Hence $\text{UA-SFE-C} \subsetneq \text{Co-SFE-C}$ under the assumption that one-way permutations exist.³¹ 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-SFE under reasonable assumptions? In the Appendix 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.

7. 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. [11], [36], [33], [37], [35] and part of [34]). While the study of Boolean functions in this model yielded clean and tight results [11], this is not the case when considering non-Boolean functions. In fact Kushilevitz [36] 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 [33] 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-to-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 $f: \{0, 1\}^{2n+1} \times \{0, 1\}^{2n} \rightarrow \{0, 1\}^{2n}$ 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-to-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.

³⁰ This disregards uniformity issues (such as how to find the insecure minor). When uniformity is taken into account we can only say that $\text{UA-SFE-C} \subseteq \text{IT-SFE-C}$.

³¹ 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 make sure that no insecure minor exists in the function f , add to the output a perfectly binding commitment of the input (see [19]). Such a commitment can be achieved non-uniformly given one-way functions.

Probabilistic Functionalities. Kilian [34] gives combinatorial criteria also for secure evaluation of probabilistic functionalities (rather than deterministic functions) in the semi-honest model. These results, however, do not necessarily yield efficient protocols thus raising the question whether computational results can be found in this model as well. While it is known that given OT one can achieve SFE of any efficiently computable functionality, it is interesting what can be achieved with other 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 OT 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 comments on this paper. We thank Ronen Shaltiel for useful advice regarding hardness amplification and Yuval Ishai for his advice on the semi-honest versus malicious models. Finally we thank the anonymous referees for their helpful comments.

Appendix. Separating UA-Eff-SFE and IT-Eff-SFE

In Example 1 we showed a function that is in IT-Eff-SFE but on the other hand in Co-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-SFE under reasonable assumptions?³² 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 let A be an assignment to its variables. Let $\Phi(A)$ denote the value of the formula Φ under the assignment A . Consider the function

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

(Alice gets no input while Bob gets input $y = (\Phi, A)$.)³³

In the semi-honest model $f_{3\text{-SAT}}$ 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:

³² Recall that functions with no insecure minor are in IT-Eff-SFE.

³³ Note that the choice of 3-SAT is immaterial and any \mathcal{NP} -complete language would have sufficed.

Claim A.1. *If there exists a malicious SFE protocol for $f_{3\text{-SAT}}$ 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 $f_{3\text{-SAT}}$ in the unbounded adversaries is actually a perfect zero knowledge proof for the following \mathcal{NP} -complete language: $3\text{-SAT} = \{\Phi \mid \Phi \text{ has a satisfying assignment}\}$. 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 $\text{co-}\mathcal{AM}$ [17], [1], [42], thus $\mathcal{NP} \subseteq \text{co-}\mathcal{AM}$. Combining this claim with a result of Boppana et al. [6] we get that $f_{3\text{-SAT}}$ does not have an SFE protocol in the malicious unbounded adversaries model unless the polynomial time hierarchy collapses to the second level. Thus $\text{UA-Eff-}\mathcal{SFE} \subsetneq \text{IT-Eff-}\mathcal{SFE}$ 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 $\text{IT-Eff-}\mathcal{SFE} = \text{UA-Eff-}\mathcal{SFE}$ unconditionally.

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

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

Also define

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

Clearly, $f_{g_0, g_1}(x, y) \in \text{IT-Eff-}\mathcal{SFE}$ 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 $f_{g_0, g_1}(x, y) \notin \text{UA-Eff-}\mathcal{SFE}$.*

Proof. Suppose that $f_{g_0, g_1}(x, y) \in \text{UA-Eff-}\mathcal{SFE}$, 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 $S_A(x, z_x) = v$. However, if $(z_0, z_1) \notin L_{g_0, g_1}$ then there exists no view (simulated or real) that corresponds to both $x = 0$ and $x = 1$ (since by the correctness of the SFE protocol Π the output must indeed be $z_x = g_x(y)$ for some x). Thus 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 $v = S_A(x, g_x(y))$. \square

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.

References

- [1] W. Aiello and J. Hastad. Statistical zero-knowledge languages can be recognized in two rounds. *Journal of Computer and Systems Sciences*, 42(3):327–345, 1991.
- [2] A. Beimel, T. Malkin, and S. Micali. The all-or-nothing nature of two-party secure computation. In *CRYPTO '99*, pages 80–97. LNCS 1666. Springer, Berlin, 1999.
- [3] M. Bellare and S. Micali. Non-interactive oblivious transfer and applications. In *CRYPTO '89*, pages 547–557. LNCS 435. Springer, Berlin, 1989.
- [4] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proc. 20th STOC*, pages 1–10, 1988.
- [5] C.H. Bennett, G. Brassard, C. Crepeau, and M.H. Skubiszewska. Practical quantum oblivious transfer protocols. In *CRYPTO '91*, pages 351–366. LNCS 576. Springer, Berlin, 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 *Proc. 39th FOCS*, pages 493–502, 1995.
- [8] R. Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000.
- [9] R. Canetti, E. Kushilevitz, and Y. Lindell. On the limitations of universally composable two-party computation without set-up assumptions. *Journal of Cryptology*, 19(2):135–167, 2006.
- [10] D. Chaum, C. Crepeau, and I. Damgard. Multiparty unconditionally secure protocols. In *Proc. 20th STOC*, pages 11–19, 1988.
- [11] B. Chor and E. Kushilevitz. A zero-one law for boolean privacy. *SIAM Journal on Discrete Mathematics*, 4(1):36–47, 1991. Preliminary version in *Proc 21st STOC*.
- [12] C. Crepeau. Equivalence between two flavours of oblivious transfers. In *CRYPTO '87*, pages 350–354. LNCS 293. Springer, Berlin, 1987.
- [13] C. Crepeau and J. Kilian. Achieving oblivious transfer using weakened security assumptions. In *Proc. 29th FOCS*, pages 42–52, 1988.
- [14] I. Damgard, J. Kilian, and L. Salvail. On the (im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions. In *Eurocrypt '99*, pages 56–73. LNCS 1592. Springer, Berlin, 1999.
- [15] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.
- [16] M. Fitzi, J.A. Garay, U.M. Maurer, and R. Ostrovsky. Minimal complete primitives for secure multi-party computation. In *CRYPTO '01*, pages 80–100. LNCS 2139. Springer, Berlin, 2001.
- [17] L. Fortnow. The complexity of perfect zero-knowledge. In *Randomness and Computation* (editor S. Micali), pages 327–343. Volume 5 of Advances in Computing Research. JAI Press, Greenwich, CT, 1989.

- [18] Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The relationship between public key encryption and oblivious transfer. In *Proc. 41st FOCS*, pages 325–335, 2000.
- [19] O. Goldreich. *Foundations of Cryptography*. Cambridge University Press, Cambridge, 2001.
- [20] O. Goldreich. *Foundations of Cryptography - Volume 2*. Cambridge University Press, Cambridge, 2004.
- [21] O. Goldreich and L.A. Levin. A hard-core predicate for all one-way functions. In *Proc. 21st STOC*, pages 25–32, 1989.
- [22] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game—a completeness theorem for protocols with honest majority. In *Proc. 19th STOC*, pages 218–229, 1987.
- [23] 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, 1991.
- [24] O. Goldreich, N. Nisan, and A. Wigderson. On Yao’s XOR-lemma. Electronic Colloquium on Computational Complexity, Report TR95-050, 1995. Available at <http://eccc.hpi-web.de/eccc/>.
- [25] O. Goldreich and R. Vainish. How to solve any protocol problem—an efficiency improvement. In *CRYPTO ’87*, pages 73–86. LNCS 293. Springer, Berlin, 1987.
- [26] M. Hirt and U. Maurer. Player simulation and general adversary structures in perfect multiparty computation. *Journal of Cryptology*, 13(1):31–60, 2000.
- [27] R. Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proc. 36th FOCS*, pages 538–545, 1995.
- [28] R. Impagliazzo. A personal view of average-case complexity. In *Proc. 10th Annual Structure in Complexity Theory Conference*, pages 134–147. IEEE Computer Society Press, Los Alamitos, CA, 1995.
- [29] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. In *Proc. 30th FOCS*, pages 230–235, 1989.
- [30] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *Proc. 21st STOC*, pages 44–61, 1989.
- [31] Y. Ishai. Personal communication, 2004.
- [32] J. Kilian. Founding cryptography on oblivious transfer. In *Proc. 20th STOC*, pages 20–31, 1988.
- [33] J. Kilian. A general completeness theorem for two-party games. In *Proc. 23rd STOC*, pages 553–560, 1991.
- [34] J. Kilian. More general completeness theorems for secure two-party computation. In *Proc. 32nd STOC*, pages 316–324, 2000.
- [35] J. Kilian, E. Kushilevitz, S. Micali, and R. Ostrovsky. Reducibility and completeness in private computations. *SIAM Journal of Computing*, 28(4):1189–1208, 2000.
- [36] E. Kushilevitz. Privacy and communication complexity. *SIAM Journal on Discrete Mathematics*, 5(2):273–284, 1992. Preliminary version in *Proc. 30th FOCS*.
- [37] E. Kushilevitz, S. Micali, and R. Ostrovsky. Reducibility and completeness in multi-party private computations. In *Proc. 35th FOCS*, pages 478–489, 1994.
- [38] L.A. Levin. One-way functions and pseudorandom generators. *Combinatorica*, 7:357–363, 1987.
- [39] U. Maurer. Information-theoretic cryptography. In *CRYPTO ’99*, pages 47–64. LNCS 1666. Springer, Berlin, 1999.
- [40] S. Micali and P. Rogaway. Secure computation. In *CRYPTO ’91*, pages 392–404. LNCS 576. Springer, Berlin, 1991.
- [41] M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *Proc. SIAM Symposium on Discrete Algorithms (SODA 2001)*, pages 448–457, 2001.
- [42] T. Okamoto. On relationships between statistical zero-knowledge proofs. *Journal of Computer and Systems Sciences*, 60(1):47–108, 2000.
- [43] R. Ostrovsky and A. Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *Second Israel Symposium on Theory of Computing Systems, ISTCS 93, Proceedings*, pages 3–17. IEEE Computer Society Press, 1993.
- [44] M. O. Rabin. How to exchange secrets by oblivious transfer. TR-81, Harvard University, Cambridge, MA, 1981.
- [45] A. C. Yao. Protocols for secure computations. In *Proc. 23rd FOCS*, pages 160–164, 1982.
- [46] A. C. Yao. Theory and application of trapdoor functions. In *Proc. 23rd FOCS*, pages 80–91, 1982.
- [47] A. C. Yao. How to generate and exchange secrets. In *Proc. 27th FOCS*, pages 162–167, 1986.