On Invertible Sampling and Adaptive Security

Yuval Ishai[1,⋆], Abishek Kumarasubramanian[2],
Claudio Orlandi[3,⋆⋆], and Amit Sahai[2,⋆⋆⋆]

[1] Technion and UCLA
yuvali@cs.technion.ac.il
[2] UCLA
{abishekk,sahai}@cs.ucla.edu
[3] Aarhus University
claudio@cs.au.dk

Abstract Secure multiparty computation (MPC) is one of the most general and well studied problems in cryptography. We focus on MPC protocols that are required to be secure even when the adversary can *adaptively* corrupt parties during the protocol, and under the assumption that honest parties cannot reliably erase their secrets prior to corruption. Previous feasibility results for adaptively secure MPC in this setting applied either to deterministic functionalities or to randomized functionalities which satisfy a certain technical requirement. The question whether adaptive security is possible for *all* functionalities was left open.
We provide the first convincing evidence that the answer to this question is negative, namely that some (randomized) functionalities cannot be realized with adaptive security.
We obtain this result by studying the following related *invertible sampling* problem: given an efficient sampling algorithm A, obtain another sampling algorithm B such that the output of B is computationally indistinguishable from the output of A, but B can be efficiently inverted (even if A cannot). This invertible sampling problem is independently motivated by other cryptographic applications. We show, under strong but well studied assumptions, that there exist efficient sampling algorithms A for which invertible sampling as above is impossible. At the same time, we show that a general feasibility result for adaptively secure MPC implies that invertible sampling is possible for every A, thereby reaching a contradiction and establishing our main negative result.

1 Introduction

Secure multiparty computation (MPC) is one of the most fundamental problems in cryptography. The goal of MPC is to allow two or more parties to compute

[⋆] Supported in part by ISF grant 1310/06, BSF grant 2008411, and NSF grants 0830803, 0716835, 0627781.

[⋆⋆] This work was done while the author was visiting UCLA.

[⋆⋆⋆] Supported in part by NSF grants 0916574, 0830803, 0716389, and 0627781, an equipment grant from Intel, and an Okawa Foundation Research Grant.

some functionality (a deterministic or randomized mapping from inputs to outputs) while emulating an ideal evaluation of the functionality in which a trusted trusted party receives all inputs and delivers all outputs. This is formally captured by simulation-based security definitions, which (roughly speaking) require that whatever an adversary can achieve by attacking the real execution of the protocol can also be achieved by a *simulator* which attacks the above ideal evaluation process.

Since the introduction of MPC in the 1980s [42,28,5,15], many security definitions have been proposed and feasibility results shown. In particular, significant research efforts have been invested in realizing *adaptively secure* MPC protocols, whose security is required to hold in the presence of an adversary that can corrupt parties *adaptively* at any point during the protocol. When considering adaptive security, it is typically assumed that honest parties cannot reliably erase their secrets. This is an assumption we make throughout this work. The main challenge in proving the security of cryptographic protocols in this setting is that when a new party is corrupted, the simulator needs to provide an explanation of the internal randomness for this party that has to be consistent with the simulated view so far and with the party's input.

Adaptively secure protocols in this setting were first constructed by Canetti, Feige, Goldreich and Naor [13] in a standalone model and then by Canetti, Lindell, Ostrovsky and Sahai [14] in the universal composability (UC) model [9]. These protocols applied to all deterministic functionalities, but in the case of randomized functionalities they were restricted to so called *adaptively well-formed* functionalities [14]. Intuitively, randomized functionalities can present the following problem: when the adversary corrupts *all* the parties in the real execution[4], he learns the private randomness of all parties. However in the ideal world, if the ideal functionality tosses some coins that are kept private and used during the computation, the ideal adversary (the simulator) will never learn these private coins, even after corrupting every party. The presence of private randomness in the ideal world makes it problematic to realize randomized functionalities in which the randomness cannot be efficiently computed from the inputs and outputs. The "adaptively well formed" functionalities satisfy the syntactic requirement that they reveal all their internal randomness when all parties are corrupted. (In other words, securely realizing such functionalities does not pose the challenge of hiding the internal randomness of the functionality from the adversary.) The question for general functionalities was left open.

In this paper we show that, under strong but well studied computational assumptions, there exist functionalities which *cannot* be realized with adaptive

[4] At first glance, it may seem strange to require any security when all parties involved in a protocol are eventually corrupted. However, this is important when protocols are meant to be composed (even sequentially). For instance, a sub-protocol of a larger protocol may involve only a small subset S of the participants of the larger protocol. In such a situation, guaranteeing security of the larger protocol when (only) the players in S are corrupted would require analyzing the security of the sub-protocol when *all* the participants of the sub-protocol are corrupted.

security. Concretely, our main negative result relies on the following two assumptions: (1) the existence of so-called *extractable one-way functions* [10,11,18] (this is a common generalization of several "knowledge-of-exponent" style assumptions from the literature [19,31,4,39]), and (2) the existence of non-interactive zero-knowledge (NIZK) proofs for NP [6,7].

Our negative result applies to almost every model of adaptively secure computation without erasures from the literature. This includes stand-alone security in the semi-honest and malicious models (under the definition of [8]), UC-security in the CRS model (under the definition of [14]) or even to security in the OT-hybrid model, where every functionality can be *unconditionally* realized with non-adaptive UC-security [35,34]. Our negative result does *not* apply to the case where only a strict subset of the parties can be corrupted (in particular, to MPC with an honest majority). The existence of uncorrupted parties allows the simulator to avoid the need for "explaining" the output of the functionality by providing its internal randomness. Our negative result also does not apply to adaptive security in the standalone model without post-execution corruption [8]; this (nonstandard) notion of adaptive security does not support even sequential composition. See Section 1.2 below.

Invertible sampling. A key concept which we use to obtain our negative result and is of independent interest is that of *invertible sampling* (Definition 1 of [20]). Suppose we are given an efficient sampling algorithm A. Can we always obtain an *alternative* efficient sampling algorithm B such that the output of B is indistinguishable from the output of A, but B can be efficiently inverted in the sense that its randomness can be efficiently computed based on its output? Here we refer to a distributional notion of inversion, namely an inversion algorithm B^{-1} is successful if the pair $(r', \mathsf{B}(r'))$ is computationally indistinguishable from the pair $(\mathsf{B}^{-1}(\mathsf{B}(r')), \mathsf{B}(r'))$ where r' is a uniform random input for B. We refer to the hypothesis that every efficient A admits an efficient B as above as the *invertible sampling hypothesis (ISH)*. While our study of ISH is primarily motivated by its relevance to adaptive security, this question is independently motivated by other cryptographic applications (such as settling the relation between public-key encryption and oblivious transfer); see Section 6 for details.

The ISH may seem easy to refute under standard assumptions. Indeed, if we require the outputs of A and B to be *identically* distributed, then ISH could be refuted based on the existence of any pseudorandom generator G: Let A output $G(r)$. The existence of B as above would allow one to distinguish between $G(r)$ (for which B^{-1} will find an inverse under B with overwhelming probability) and a uniformly random string of the same length (which with overwhelming probability has no inverse under B). However, the case where the outputs of B and A should only be *computationally* indistinguishable appears to be much more challenging. In particular, note that a pseudorandom distribution *does* admit an invertible alternative sampler: the sampler B just outputs a uniformly random string. Since this output is computationally indistinguishable from the actual distribution, it is consistent with the above formulation of ISH.

We show, under the assumptions described above, that there exist efficient sampling algorithms A for which the ISH fails. At the same time, we show that a general feasibility result for adaptively secure MPC implies that invertible sampling is possible for every A, thereby reaching a contradiction and establishing our main negative result.

More precisely, we show that general adaptively secure computation implies (and in fact, is equivalent to) a stronger version of ISH in which the sampling algorithms A, B are given an input x in addition to their random input, and where the inversion algorithm B^{-1} should be successful on *every* input x. This stronger flavor of ISH is ruled out by the assumptions mentioned above, namely the existence of extractable one-way functions and NIZK proof systems for NP. To rule out the weaker variant of ISH (with no input x) we need to use somewhat stronger assumptions: a non-standard (but still plausible) variant of an extractable one-way function, and the existence of non-interactive witness-indistinguishable (NIWI) protocols for NP without a common reference string [22,1,29,30].

1.1 Our Techniques

We now give some intuition on our construction of an efficient sampling algorithm A for which ISH does not hold. For this purpose, it is convenient to first describe a *relativized* world (defined via a randomized oracle) in which such A provably exists. As a first attempt, suppose that we have an oracle computing a random function $f : \{0,1\}^n \to \{0,1\}^{2n}$. Now, consider the efficient sampling algorithm A which outputs a random image of f, namely $\mathsf{A}(r) = f(r)$. (Note that A is efficient given oracle access to f.) Similarly to the previous PRG example, such an algorithm is not enough to refute the computational version of ISH: indeed, the alternative sampler B can simply output a uniformly random string of length $2n$. The high level idea for ruling out such an alternative sampler is to make the outputs of f efficiently verifiable. Formally, we add to f an additional oracle g which decides whether a given string $y \in \{0,1\}^{2n}$ is in the image of f. (A similar oracle was used by Wee [41] in the seemingly unrelated context of separating two notions of computational entropy.)

We now informally argue that ISH is false relative to the randomized oracle (f, g). Let $\mathsf{A}(r) = f(r)$ as before. Assume towards a contradiction that an alternative sampling algorithm $\mathsf{B}(r')$ as required by ISH exists. We argue that B can be used to efficiently invert f on a random output $y = f(x)$, which remains hard even when given the decision oracle g. By the computational indistinguishability requirement, it suffices (in order to reach a contradiction) to successfully invert f on a random output y' sampled by B. Moreover, since indistinguishability holds relative to the verification oracle g we are guaranteed that (with overwhelming probability) y' as above will be in the image of f.

The inversion algorithm for f, when given y' sampled by B, uses the inversion algorithm B^{-1} guaranteed by ISH to obtain a preimage r' of y' under B. Since f is a random function, it is impossible to efficiently find an image y' of f without querying f on the corresponding pre-image. (Jumping ahead, this is the step where our explicit non-relativized construction will rely on a knowledge assumption.) Thus, the inversion algorithm can use r' to extract a preimage x of y' under f by running B on r' until it queries f on a point x such that $f(x) = y$.

To obtain an explicit version of the above A we use extractable one-way functions to implement f and a NIZK proof for proving range membership to emulate the role of g (the latter is similar to the use of NIZK proofs in [37,32]; see Section 1.2). For technical reasons that have to do with the common reference string required by the NIZK proof system, we cannot use this approach to refute the basic version of ISH described above. For this, we need to employ a somewhat more complicated proof strategy and apply NIWI proofs instead of NIZK proofs. See Section 4 for details.

1.2 Related Work

Adaptively secure MPC (without erasures) was first realized in [13] for the stand-alone case. In [8], a variant of the notion of adaptive security that guarantees sequential composition was introduced: we refer to the variant from [8] which requires security against *post execution corruption (PEC)*. Namely, after the simulation is complete, the environment can ask the adversary to corrupt additional parties and simulate their views. This variant is used in [8] to prove sequential composition. In fact, a separation between adaptive security with PEC and without it has been shown in [12]. We stress that the negative results from [12] apply to specific *protocols* rather than functionalities. That is, [12] builds protocols which are shown to be adaptively secure in one setting but not adaptively secure in another setting, but does not show any *functionality* which cannot be realized with adaptive security, as opposed to our impossibility result.

In the UC security framework [9] the main feasibility result for securely realizing adaptively well-formed functionalities against an adaptive adversary was obtained in [14] (see also [17,25]). This work also suggested the following plausible candidate for a randomized functionality which *cannot* be realized with adaptive security: on input a security parameter k, output the product of two random k-bit primes. However, we do not know how to relate the possibility of realizing this functionality with adaptive security to any well-studied assumption.

If one is willing to assume that honest parties can reliably erase their data, security against adaptive adversaries becomes a much easier task. Our negative results do not apply to this alternative model, and general feasibility results in this model were obtained in [3,36].

The Invertible Sampling Hypothesis is related to questions of oblivious sampling that have been studied in other cryptographic contexts. For instance, the question of generating a public key for an encryption scheme without learning how to decrypt is related to the goal of constructing an oblivious transfer protocol from a public-key encryption scheme [23,26]; virtually any non-committing encryption scheme [2,20,25,16] requires some form of oblivious sampling of public keys; in a recent result [21] the question of whether ISH holds has been informally asked, in the context of turning UC-secure protocols in the common reference string model into semi-honest secure stand-alone protocols. If the common reference string is a random string, the problem trivially reduces to having one party publish a random string. If the CRS instead is sampled using some generic distribution, is not clear whether a semi-honest party can sample the common reference string without learning the trapdoor. See Section 6 for a further discussion of these additional connections of ISH with cryptography.

The *knowledge of exponent assumption* was introduced in [19], and since then other specific knowledge of exponent assumptions have been proposed [4,39], until in some recent work [11,10,18] the abstract notion of *extractable functions* has been introduced. Our impossibility results rely on assumptions of this type. The use of knowledge assumptions in security proofs has received criticism in the cryptographic community, especially because such assumptions seem hard to disprove [38] (even though in [4] a "wrong" knowledge assumption from [31] has been disproved). As far as we know, our work is the first to apply such assumptions towards *negative* results in cryptography.

Finally, our use of NIZK and NIWI proofs for NP was inspired by the use of NIZK in [37] to construct a class of distributions where efficient learning with an evaluator is possible but coming up with a generator that approximates the given distribution is infeasible, and by [41,32] in the context of separating conditional HILL and Yao entropies. Note, however, that none of these works made use of knowledge assumptions; such assumptions appear to be crucial to our techniques.

2 Preliminaries

Notation. We use n as a length parameter; all probability distributions we consider in this work will be over strings of length polynomial in n. We let U_n denote the uniform distribution over $\{0,1\}^n$. We use $x \leftarrow X$ to denote the process of sampling x from the distribution X. If X is a set, $x \leftarrow X$ denotes a uniform choice of x from X. For any distribution X and algorithm A, we denote by $\mathsf{A}(X)$ the probability distribution on the outputs of A taken over the coin tosses (if any) of A and an independent random choice of the input x from X.

We use the standard notation $\{C_1; C_2; \ldots; C_m : D\}$ to denote the distribution of D obtained as a result of the sampling process defined by the sequence of instructions $C_1, \ldots, C_m$. For example, $\{a \leftarrow X; b \leftarrow \mathsf{A}(a) : (a, b)\}$ denotes the distribution of pairs (a, b) obtained by first picking a from X and then obtaining b by running A on a. Similarly, we use $\Pr[C_1; C_2; \ldots; C_m : E]$ to denote the probability of event E in the probability space defined by the sequence of instructions $C_1, \ldots, C_m$. For instance, $\Pr[a \leftarrow X; b \leftarrow Y : a \neq b]$ is the probability that when a is chosen according to X and b is independently chosen according to Y, a and b are not equal.

We assume that the reader is familiar with the concepts of *negligible function*, *one-way function*, *pseudorandom generator*, and *non-interactive zero-knowledge proof system*. Suitable definitions can be found in the full version or in [27].

By default we assume efficient algorithms to be uniform and efficient distinguishers to be nonuniform. We will use $\epsilon(\cdot)$ to denote an unspecified negligible function.

Let $I \subseteq \{0,1\}^*$ be an arbitrary infinite index set. We say that two distribution ensembles $\{X_w\}_{w\in I}$ and $\{Y_w\}_{w\in I}$ are *computationally indistinguishable* if for every polynomial-size circuit family C_n there exists a negligible function ϵ such that for every $w \in I$,

$$|\Pr[C_{|w|}(X_w) = 1] - \Pr[C_{|w|}(Y_w) = 1]| \leq \epsilon(|w|).$$

Sampling algorithms. We will view any probabilistic polynomial time (PPT) algorithm A as defining an *efficient sampling algorithm* (or *sampler* for short). We let $\mathsf{A}(w)$ denote the output distribution of A on input w and $\mathsf{A}(w; r_{\mathsf{A}})$ denote the output when the random input (i.e., sequence of coin-tosses) is given by r_{A}. Without loss of generality, we can associate with every efficient A a polynomial $\ell(\cdot)$ such that r_{A} is a random input of length $\ell(|w|)$. Under this convention, $\mathsf{A}(w)$ is distributed identically to $\mathsf{A}(w; U_{\ell(|w|)})$. We will use this convention throughout the paper. Finally, we will sometimes be interested in the special case of samplers over a unary input alphabet; in this case A defines a sequence of distributions $\{\mathsf{A}(1^n)\}_{n\in\mathbb{N}}$.

We say that a sampling algorithm A is *inverse-samplable* if there exists a PPT inversion algorithm which, given an input w and a sample y from the output $\mathsf{A}(w)$, outputs a random input r for A which is consistent with w, y. Moreover, the choice of r should be "correctly distributed" in the sense that (w, y, r) should be computationally indistinguishable from $(w, \mathsf{A}(r_{\mathsf{A}}), r_{\mathsf{A}})$ where $r_{\mathsf{A}} \leftarrow U_{\ell(|w|)}$. (Such a distributional inversion requirement is similar in spirit to the definition of a distributionally one-way function [33].)

Definition 1. (Inverse-Samplable Algorithm) *We say that an efficient sampling algorithm* A *is* inverse-samplable *if there exists a PPT* inverter *algorithm* A^{-1} *such that the distribution ensembles* $\{r_{\mathsf{A}} \leftarrow U_{\ell(|w|)} : (r_{\mathsf{A}}, \mathsf{A}(w; r_{\mathsf{A}}))\}_{w\in\{0,1\}^*}$ *and* $\{r_{\mathsf{A}} \leftarrow U_{\ell(|w|)} : (\mathsf{A}^{-1}(w, \mathsf{A}(w; r_{\mathsf{A}})), \mathsf{A}(w; r_{\mathsf{A}}))\}_{w\in\{0,1\}^*}$ *are computationally indistinguishable.*

3 Invertible Sampling Hypothesis

The Invertible Sampling Hypothesis (ISH) is concerned with the possibility of inverse-sampling arbitrary efficiently samplable distributions. It is easy to see that if one-way functions exist, then there are efficient sampling algorithms which are not inverse-samplable. Thus, we settle for the hope that for every efficient sampling algorithm A there exists an efficient and inverse-samplable algorithm B whose output is computationally indistinguishable from that of A. The ISH captures the above hope. We will also consider a weaker variant of ISH, referred to as *weak ISH*, which restricts the sampler A to have a unary input alphabet. This is formalized below.

Hypothesis 1. (Invertible Sampling Hypothesis: ISH) *For every efficient sampling algorithm* A *there exists an efficient sampling algorithm* B *satisfying the following two requirements.*

1. Closeness: *The distribution ensembles* $\{\mathsf{A}(w)\}_{w\in\{0,1\}^*}$ *and* $\{\mathsf{B}(w)\}_{w\in\{0,1\}^*}$ *are computationally indistinguishable.*
2. Invertibility: B *is inverse-samplable (see Definition 1).*

Hypothesis 2. (Weak ISH) *The* weak ISH *is defined exactly as ISH above, except that the inputs* w *for* A *and* B *are restricted to be unary (i.e., are of the form* 1^n*).*

Clearly, ISH implies Weak ISH. The weaker flavor of ISH is somewhat more natural in that it refers to the traditional notion of a sampling algorithm (defining a single probability distribution for each length parameter n) as opposed to the more general notion of a probabilistic algorithm. Moreover, the weak ISH suffices for the motivating applications discussed in Section 6. However, it turns out that the stronger flavor can be refuted under more standard assumptions and that ruling out this flavor suffices for obtaining our main negative result on adaptively secure MPC. Thus, in the following we will consider both variants of ISH.

We will start (in Section 4) by refuting the weak ISH assuming the existence of a strong variant of extractable one-way functions as well as NIWI proof systems for NP. We will then (Section 5) refute the original and stronger variant of ISH under the weaker assumptions that standard extractable one-way functions (generalizing various "knowledge-of-exponent assumptions" from the literature) exist, as well as NIZK protocols for NP in the CRS model. At a high level, refuting the stronger flavor of ISH is easier because the additional "external" input allows us to introduce randomness over which the alternative sampler B has no control. This randomness can be used for choosing the CRS for a NIZK proof or random parameters for a family of extractable one-way functions.

4 Conditional Refutation of Weak ISH

As already discussed in the introduction, any pseudorandom generator $G : \{0,1\}^n \to \{0,1\}^{2n}$ provides a nontrivial example of a sampling algorithm for which weak ISH holds. Indeed, if $\mathsf{A}(1^n)$ outputs $G(r_\mathsf{A})$ where $r_\mathsf{A} \leftarrow U_n$, then $\mathsf{B}(1^n)$ can simply output r_B where $r_\mathsf{B} \leftarrow U_{2n}$.

This example suggests that in order to provide a counterexample for the (weak) ISH, it does not suffice for the computation performed by the sampler to be *one-way* and for its output support to be *sparse*, but its output should also be *verifiable* (a feature missing in the aforementioned example). Jumping ahead, verifiability will be achieved via variants of non-interactive zero-knowledge. It turns out that even the "sparseness" requirement needs to be significantly strengthened in order to rule out the possibility of directly sampling an output without knowing a corresponding input. Classes of sparse one-way functions with a similar property were studied in [19,4,39] under the umbrella of "knowledge assumptions." Crudely speaking, a knowledge assumption for a function f states that if any efficient algorithm A outputs a point in $\mathsf{image}(f)$, then the only way A could have computed this image is by choosing an x and computing $f(x)$ (here it is necessary that $\mathsf{image}(f)$ be sparse). Thus the algorithm "knows" x. This is formally captured by requiring the existence of an efficient algorithm that can extract x from A's input and randomness.

A brief outline of our refutation of weak ISH is as follows. Suppose a function f is both "extractable" and one-way. Given an algorithm which produces valid points in $\mathsf{image}(f)$, if we can obtain the randomness that it used, then we can use f's "knowledge extractor" to find pre-images and thus break the one-wayness of f. However to obtain this randomness, we need the algorithm to be inverse-samplable. Since weak ISH hypothesizes the existence of such an algorithm we can invert f and contradict its one-wayness.

Next, we formally prove that weak ISH is false assuming the existence of a strong notion of an Extractable One-Way Function (EOWF) and the assumption that Non-Interactive Witness Indistinguishable Proofs (NIWI) exist for all of NP.

We start by defining the two primitives we rely on. An extractable one-way function is a one-way function f with the following extraction property: for any efficient A which, on random input r_A, attempts to output an element y in the image of f, there is an efficient extractor K_A which given the random input r_A of A succeeds in finding a preimage $x \in f^{-1}(y)$ with roughly the same success probability. Formally:

Definition 2. (Extractable One-Way Function (EOWF)) *Let f be a one-way function. We say that f is an* extractable one-way function *if for every PPT algorithm A with running time $\ell(n)$ there is a PPT extractor algorithm K_A such that for every n:*

$$Pr[r_A \leftarrow U_{\ell(n)}; y = A(1^n; r_A); x \leftarrow K_A(1^n, r_A) : \\ (f(x) = y) \vee (\forall x', f(x') \neq y)] \geq 1 - \epsilon(n)$$

for some negligible function ϵ.

We note that the above definition appears stronger than similar definitions from the literature in that it requires f to be a single, explicit one-way function, as opposed to a keyed collection of functions. In particular, EOWF as above *can not* be instantiated using concrete knowledge assumptions from the literature such as the ones in [19,4,39]. However, it still seems plausible that (length-flexible versions of) practical cryptographic functions satisfy the above definition. In Section 5 we will rely on a more standard notion of EOWF (which allows f to depend on a random key and captures previous assumptions from the literature) in order to refute the strong variant of ISH.

Next we need the notion of *non-interactive witness indistinguishable (NIWI)* proof systems [1,29,30]. A NIWI proof is used to efficiently prove that an input x is in some NP-language L without allowing the verifier to distinguish between any two possible witnesses. While the latter witness indistinguishability property is weaker than the zero-knowledge property of NIZKs, it turns out that it is sufficient for our purposes. The important advantage of NIWI proofs is that they can be implemented (under stronger assumptions) without a trusted common reference string, which is inherently required for NIZK proofs.

Definition 3. (Non-Interactive Witness Indistinguishable Proof System [22,1,29]) *Let L be any NP language, and R_L a fixed witness relation for L. Then $\mathcal{P} = (P, V)$ is called a* non-interactive witness indistinguishable (NIWI) proof system *for R_L if P and V are PPT algorithms and the following conditions hold for some negligible function ϵ:*

1. Completeness. *For all $(x, w) \in R_L$*
$$\Pr[\pi \leftarrow P(x, w); b \leftarrow V(x, \pi) : b = 1] \geq 1 - \epsilon(|x|).$$
2. Soundness. *For all $x \notin L$, for all proof strings π^**
$$\Pr[V(x, \pi^*) = 1] \leq \epsilon(|x|).$$
3. Witness Indistinguishability (WI). *For every polynomial-size circuit family C_n, and every x, w_0, w_1 such that $(x, w_0) \in R_L$ and $(x, w_1) \in R_L$,*
$$|\Pr[C_{|x|}(P(x, w_0)) = 1] - \Pr[C_{|x|}(P(x, w_1)) = 1]| \leq \epsilon(|x|).$$

NIWI proofs exist for all of NP under well-studied assumptions [22,1,29].

We now use the above two primitives to establish the main result of this section.

Theorem 1. *If EOWF exists and NIWI proofs exist for NP, then Weak ISH is false.*

Proof (sketch): Let f be an EOWF. We first define an efficient sampling algorithm A, which outputs two random points in $\mathsf{image}(f)$ and also a NIWI proof that at least one of the points was correctly computed. That is, the sampling algorithm picks random $x_0, x_1 \leftarrow \{0,1\}^n$ and outputs $(f(x_0), f(x_1), \pi)$, where π is a NIWI proof that *either* $f(x_0)$ or $f(x_1)$ is in the image of f. More concretely, π is obtained by running a NIWI prover for the NP relation defined by

$$R_L((y_0, y_1), w) = 1 \text{ iff } f(w) = y_0 \vee f(w) = y_1$$

on input $(f(x_0), f(x_1))$ and witness x_0. From Weak ISH, we obtain A's invertible alternate sampling algorithm B and its inverter B^{-1}. By the soundness property of the NIWI proof, we are (essentially) ensured that the alternate sampler B outputs at least one valid point in the image of f. But then we can construct a new algorithm X that runs B and outputs at random one of the two images y_b.

Now X is an algorithm that outputs (with significant probability) valid points in the image of f. Given that f is an EOWF, there must exist also an extractor K_{X} that given the random input of X outputs x_b such that $y_b = f(x_b)$. Using B^{-1} to inverse-sample the random input of X and feeding it to K_{X} we can efficiently invert f, contradicting its one-wayness. See the full version for more details. □

5 Conditional Refutation of ISH

In this section we refute the main (strong) variant of ISH under weaker and more standard assumptions than those used to refute Weak ISH.

We start by defining a relaxed notion of extractable one-way function which is similar to the notion of (non-interactively) extractable function family ensemble put forward by Canetti and Dakdouk [10,11,18]. In contrast to the previous notion from Definition 2, the relaxed notion follows from previous concrete knowledge assumptions in the literature such as Damgård's *knowledge of exponent* assumption [19].

Definition 4. (Function Family Ensemble) *A* function family*, indexed by a key space K, is a set of functions $F = \{f_k\}_{k \in K}$ in which each function has the same domain and range. A* function family ensemble*, $\mathcal{F} = \{F^n\}_{n \in \mathbb{N}}$, is defined as an ensemble of function families F_n with key spaces $\{K_n\}_{n \in \mathbb{N}}$.*

Definition 5. (One-Way Function Family Ensemble) *A function family ensemble is* one-way *if:*

- *f_k can be evaluated (given 1^n, $k \in K_n$, and $x \in domain(f_k)$) in time polynomial in n, and*
- *for every polynomial-size circuit family C_n there is a negligible function ϵ such that for every n,*

$$Pr[k \leftarrow K_n; x \leftarrow domain(f_k); x' = C_n(1^n, k, f_k(x)) : f_k(x') = f_k(x)] \leq \epsilon(n).$$

Definition 6. (Non-Interactively Extractable One-Way Function Family Ensembles [18]) *We say that an one-way function family ensemble is* non-interactively extractable *(without auxiliary information) if for any efficient sampling algorithm A running in time $\ell(n)$ (with random input $r_A \in U_{\ell(n)}$), there exists a PPT algorithm K_A and a negligible function ϵ such that for all n:*

$$Pr[k \leftarrow K_n; r_A \leftarrow U_{\ell(n)}; y = A(1^n, k; r_A); x \leftarrow K_A(1^n, k, r_A) : \\ (f_k(x) = y) \vee (\forall x', f_k(x') \neq y)] \geq 1 - \epsilon(n).$$

The difference between the above notion of extractable one-way function family ensembles and the notion of EOWF from Definition 2 is that extraction is not guaranteed for all functions in the function family but only for a randomly chosen function (concretely, the first step $k \leftarrow K_n$ chooses a random function). Furthermore, the process of picking the random function may use private randomness that is not available to the algorithm A.

The above difference makes it possible to derive extractable one-way function family ensembles from existing knowledge assumptions in literature [19,31,4,39]. As an example, the Knowledge of Exponent (KEA) Assumption [19] informally states that there exists an ensemble of groups $\{G_n\}_{n \in \mathbb{N}}$ where the discrete logarithm problem is hard to solve and any PPT adversary A that on input g, w can compute a pair of the form (g^r, w^r) must *know* r, in the sense that there exists an efficient extractor K_A which given the random input of A can compute r. Mapping this example to Definition 6, the key space is $K_n = G_n \times G_n$ and the function f_k with $k = (w, g)$ is defined by $f_k(r) = (w^r, g^r)$.

Next, we replace the previous NIWI primitive with non-interactive zero knowledge (NIZK) proofs in the common reference string (CRS) model [6,40]. We omit the (standard) definition of NIZK, but note that the assumptions on which NIZK proof systems for NP can be based are significantly more general than the corresponding assumptions for NIWI, and include the existence of trapdoor permutations [24].

We are now ready to state the main theorem of this section.

Theorem 2. *If non-interactively extractable one-way function family ensembles exist and NIZK proof systems exist for NP, then ISH is false.*

Proof (sketch): The proof follows the same outline as the one from Theorem 1, but the use of NIZK instead of NIWI allows it to take a somewhat simpler form. Let $\mathcal{F}$ be a non-interactively extractable one-way function family ensemble. We first define an efficient sampling algorithm A whose inputs are pairs of strings (k, σ): k is a key from the key space of $\mathcal{F}$ and σ is a uniformly random string to be used as a CRS for a NIZK proof system. A outputs a random image of f_k and a NIZK proof (under σ) that the output is valid. Let B be the alternate invertible sampler hypothesized by ISH. Due to the soundness of the NIZK proof system, B outputs valid images of f_k when σ is chosen uniformly at random. Since $\mathcal{F}$ is extractable, we can use B, its extractor K_{B} and its inverter B^{-1} to construct an efficient inversion algorithm for the family ensemble $\mathcal{F}$, contradicting its one-wayness property. See the full version for details. □

6 Applications of ISH

While our main motivation for studying ISH is its relevance to adaptively secure MPC (discussed later in Section 7) we start by presenting two other consequences of (weak) ISH. In order to avoid any confusion, we remind the reader that in the previous sections we disproved ISH under some specific computational assumptions. However, as we couldn't disprove ISH *unconditionally* (or even under standard cryptographic assumptions), it is still interesting to investigate the consequences of ISH in order to put ISH in the proper cryptographic context and to further motivate our study.

PKE and OT: As a first consequence, we note that if ISH holds, this would settle the question of the relationship between public key encryption (PKE) and oblivious transfer (OT), as studied in [26].

Theorem 3. *If ISH holds, then the existence of semantic secure PKE implies the existence of an oblivious transfer protocol.*

Proof (sketch): The proof follows by considering a protocol for $\frac{1}{2}$-OT similar in spirit to the EGL protocol [23], where the receiver samples one public key with the key generation algorithm (thus learning the secret key), and the other using the alternate inverse-samplable algorithm, as described in ISH. Receiver's security loosely follows from the closeness property of ISH, while sender security can be deduced by the semantic security of the PKE scheme. See the full version for details. □

Assumptions for UC-secure computation: A systematic study of the minimal setup and computational assumptions for UC-secure computation has been recently undertaken in [21]. A question that the authors left open is whether the existence of stand-alone oblivious transfer (SA-OT) is a necessary assumption for UC-secure oblivious transfer (UC-OT) in the common reference string (CRS) model, where the string is sampled from an arbitrary distribution. If ISH holds, one could answer this question affirmatively. To show that SA-OT is necessary for UC-OT we will show how to construct a protocol for SA-OT assuming that UC-OT in the CRS model exists. Intuitively we need to generate a CRS to make the protocol work, but we don't want any party to learn the corresponding trapdoor. Unfortunately, we cannot let the parties use MPC in order to generate this CRS, since unconditional MPC is impossible, and we cannot assume that OT exists (or any equivalent computational assumption). But if ISH holds, there is a way of sampling any CRS without learning the trapdoor by using the invertible sampler, after which parties can run the UC-OT with respect to this CRS. Also note that we don't need this fake CRS to be distributed exactly as the real CRS, but just computationally close: if the UC-OT protocol works with the real CRS but not with the fake CRS, it could be used as a distinguisher, thus violating ISH. Standard compilation techniques can be used to turn this protocol into a protocol secure against a malicious adversary.

7 Adaptive Security and ISH

In this section we show that our strong variant of ISH (Hypothesis 1) is closely related to secure multi-party computation with security against adaptive adversaries (*adaptive MPC* or *AMPC* for short). We first show that if *all* randomized functionalities admit AMPC protocols, then ISH is true. Combined with Theorem 2, this gives the first strong evidence that *general* AMPC is impossible. Then, we proceed to show that if ISH is true and all the parties are mutually connected with OT-channels,[5] then general AMPC is possible – thus showing that ISH is essentially equivalent to general AMPC.

As discussed in the introduction, our results apply to a wide range of AMPC models from the literature. For convenience, we will refer to the two-party semi-honest model, under the definition of [8] which requires security against *post*

[5] Our use of ideal OT can be replaced by any adaptively secure OT protocol, which can be based on standard cryptographic assumptions.

execution corruption (PEC). The latter means that after the execution is complete, the environment can ask the adversary to corrupt additional parties. The PEC requirement is needed to prove sequential composition of adaptively secure protocols, and is implied by most other definitions of adaptive security from the literature (such as adaptive UC-security). Our negative result does not hold for adaptively secure protocols without PEC (since in the semi-honest two-party case, security in this model is equivalent to non-adaptive security [12]).

Brief Preliminaries. This section is an informal introduction to (adaptively secure) MPC protocols. In an MPC protocol, *adaptive security* implies that an adversarial entity can adaptively choose the parties he wants to corrupt at any point in the protocol. An adversary is *semi-honest* if the parties that he corrupts always follow the prescribed protocol. His goal is to try and obtain as much information as possible under this constraint. Security against such adversaries is a basic requirement for any cryptographic protocol. An ideal model of security for MPC protocols is one in which there exists a trusted third party who (via secure private channels) receives all the inputs from the participants of the protocol and sends back their respective outputs. Semi-honest adversaries in this model can only learn the input and output of the parties that he corrupts. Considering this as a basis for security, in the ideal-real model of [8], a real world protocol for MPC is secure if for every adversary A in the real execution, there exists an ideal world adversary S (also known as the simulator), such that the outputs of A and S are computationally indistinguishable. We refer the reader to [8,36] for a more precise definition of this notion.

7.1 Adaptively Secure MPC Implies ISH

First we show that if AMPC protocols exist for every functionality $\mathcal{F}$, then ISH (Hypothesis 1) is true.

Theorem 4. *If for every PPT functionality $\mathcal{F}$ there exists a protocol Π that securely realizes $\mathcal{F}$ against an adaptive semi-honest adversary (with PEC), then ISH is true.*

Proof (sketch): Consider a two-party randomized functionality $\mathcal{F}$ that takes input from both parties and uses some internal coins and compute some function A. Now if there exist a protocol π between P_1, P_2 that securely implements $\mathcal{F}$, in particular the following two conditions will be satisfied: 1) The output of the protocol π and the functionality $\mathcal{F}$ are computationally close (because the protocol is *correct*); 2) There exist a simulator S that can explain the randomness used by P_1, P_2 in π to produce the output z, without access to the functionality random tape $r_{\mathcal{F}}$. Therefore we can use the protocol and the simulator (π, S) as a foundation to build the inverse-samplable algorithm $\mathsf{B}, \mathsf{B}^{-1}$ that satisfy the requirement of ISH. The inverse-samplable algorithm B can be constructed by simulating a run of the protocol π between P_1 and P_2 "in the head", while the inverter B^{-1} will run the simulator S as a subroutine. See the full version for more details. □

7.2 ISH Implies Adaptively Secure MPC

In the previous section we showed that AMPC implies ISH. Now we show that the converse is true too.

To make the result stronger, we will show that ISH implies the strongest variant of MPC i.e., multiparty computation secure against an active, adaptive adversary in the universally composable security framework (UC-AMPC). Given that UC computation is impossible in plain model, we look at the OT-hybrid model where it is possible to evaluate adaptively well-formed functionalities [14], and we show how ISH would allow us to extend this result to all functionalities. We refer the reader to [9] for the definition of UC-AMPC. We look at adaptive security, where the adversary $\mathcal{A}$ can corrupt any of the two parties P_1, P_2 at any point during the protocol π.

Theorem 5. *If ISH holds, then active secure UC-AMPC is possible for any functionality in the UC-OT hybrid model.*

Proof (sketch): It is known that any deterministic functionality can be securely implemented in the OT-hybrid model[35,34]. Using the UC composition theorem and ISH we extend the result for randomized functionalities.

Consider a general randomized functionality $(z_1, z_2) \leftarrow \mathcal{F}(x, y; \rho)$, where ρ is the private randomness of $\mathcal{F}$, (x, z_1) the input/output of P_1, and (y, z_2) the input/output of P_2. Let $z_i = f_i(x, y; \rho)$. Then from Strong ISH we know that there exist f_i', f_i^{-1}, the alternative sampler and the inverter.

Now define a new, deterministic functionality $\mathcal{G}$ as $(z_1, z_2) = \mathcal{G}((x, \rho_1), (y, \rho_2))$, where $z_i = f_i'(x, y; \rho_1^i \oplus \rho_2^i)$, and where f_i' is the alternative sampler for f_i. Being a deterministic functionality, $\mathcal{G}$ can be securely realized with adaptive security in the OT-hybrid model.

Now the protocol to implement $\mathcal{F}$ in the $\mathcal{G}$-hybrid model proceeds as follows. Party P_i picks ρ_i at random, feeds it into $\mathcal{G}$ together with its input, and waits to receive the output. Note that the protocol does not exactly compute the required functionality f, but f'. The indistinguishability requirements of ISH imply that the output of f and of f' are indistinguishable too, and that suffices for UC-computation. This protocol can be shown to be UC-secure, see the full version for more details. □

References

1. Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. *SIAM J. Comput.*, 37(2):380–400, 2007.
2. Donald Beaver. Plug and play encryption. In Burton S. Kaliski Jr., editor, *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 75–89. Springer, 1997.
3. Donald Beaver and Stuart Haber. Cryptographic protocols provably secure against dynamic adversaries. In *EUROCRYPT*, pages 307–323, 1992.

4. Mihir Bellare and Adriana Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In Matthew K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 273–289. Springer, 2004.
5. Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *STOC*, pages 1–10, 1988.
6. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC*, pages 103–112, 1988.
7. Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Noninteractive zero-knowledge. *SIAM J. Comput.*, 20(6):1084–1118, 1991.
8. Ran Canetti. Security and composition of multiparty cryptographic protocols. *J. Cryptology*, 13(1):143–202, 2000.
9. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*, pages 136–145, 2001.
10. Ran Canetti and Ronny Ramzi Dakdouk. Extractable perfectly one-way functions. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP (2)*, volume 5126 of *Lecture Notes in Computer Science*, pages 449–460. Springer, 2008.
11. Ran Canetti and Ronny Ramzi Dakdouk. Towards a theory of extractable functions. In Omer Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 595–613. Springer, 2009.
12. Ran Canetti, Ivan Damgård, Stefan Dziembowski, Yuval Ishai, and Tal Malkin. Adaptive versus non-adaptive security of multi-party protocols. *J. Cryptology*, 17(3):153–207, 2004.
13. Ran Canetti, Uriel Feige, Oded Goldreich, and Moni Naor. Adaptively secure multi-party computation. In *STOC*, pages 639–648, 1996.
14. Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *STOC*, pages 494–503, 2002.
15. David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In *STOC*, pages 11–19, 1988.
16. Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Improved non-committing encryption with applications to adaptively secure protocols. In Mitsuru Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 287–302. Springer, 2009.
17. Seung Geol Choi, Dana Dachman Soled, Tal Malkin, and Hoeteck Wee. Simple, black-box constructions of adaptively secure protocols. In *TCC*, pages 387–402, 2009.
18. Ronny Ramzi Dakdouk. Theory and application of extractable functions. In *http://cs-www.cs.yale.edu/homes/jf/Ronny-thesis.pdf*, 2009.
19. Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 445–456. Springer, 1991.
20. Ivan Damgård and Jesper Buus Nielsen. Improved non-committing encryption schemes based on a general complexity assumption. In Mihir Bellare, editor, *CRYPTO*, volume 1880 of *Lecture Notes in Computer Science*, pages 432–450. Springer, 2000.
21. Ivan Damgård, Jesper Buus Nielsen, and Claudio Orlandi. On the necessary and sufficient assumptions for UC computation. In Daniele Micciancio, editor, *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 109–127. Springer, 2010.

22. Cynthia Dwork and Moni Naor. Zaps and their applications. *SIAM J. Comput.*, 36(6):1513–1543, 2007.
23. Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985.
24. Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *FOCS*, pages 308–317, 1990.
25. Juan A. Garay, Daniel Wichs, and Hong-Sheng Zhou. Somewhat non-committing encryption and efficient adaptively secure oblivious transfer. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 505–523. Springer, 2009.
26. Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *FOCS*, pages 325–335, 2000.
27. O. Goldreich. *Foundations of cryptography: Basic applications*. Cambridge Univ Pr, 2004.
28. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. pages 218–229, 1987.
29. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive Zaps and new techniques for NIZK. In *CRYPTO*, pages 97–111, 2006.
30. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432. Springer, 2008.
31. Satoshi Hada and Toshiaki Tanaka. A relationship between one-wayness and correlation intractability. In *Public Key Cryptography*, pages 82–96, 1999.
32. Chun-Yuan Hsiao, Chi-Jen Lu, and Leonid Reyzin. Conditional computational entropy, or toward separating pseudoentropy from compressibility. In Moni Naor, editor, *EUROCRYPT*, volume 4515 of *Lecture Notes in Computer Science*, pages 169–186. Springer, 2007.
33. Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *FOCS*, pages 230–235, 1989.
34. Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In David Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 572–591. Springer, 2008.
35. Joe Kilian. Founding cryptography on oblivious transfer. In *STOC*, pages 20–31, 1988.
36. Andrew Y. Lindell. Adaptively secure two-party computation with erasures. In *CT-RSA*, pages 117–132, 2009.
37. Moni Naor. Evaluation may be easier than generation (extended abstract). In *STOC*, pages 74–83, 1996.
38. Moni Naor. On cryptographic assumptions and challenges. In *CRYPTO*, pages 96–109, 2003.
39. Manoj Prabhakaran and Rui Xue. Statistically hiding sets. In *CT-RSA*, pages 100–116, 2009.
40. Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Non-interactive zero-knowledge proof systems. In *CRYPTO*, pages 52–72, 1987.
41. Hoeteck Wee. On pseudoentropy versus compressibility. In *IEEE Conference on Computational Complexity*, pages 29–41, 2004.
42. Andrew Chi-Chih Yao. How to generate and exchange secrets. pages 162–167, 1986.