On the Power of Public-key Encryption in Secure Computation

Mohammad Mahmoody[*] Hemanta K. Maji[†] Manoj Prabhakaran[‡]

September 28, 2013

Abstract

We qualitatively separate semi-honest secure computation of non-trivial secure-function evaluation (SFE) functionalities from existence of key-agreement protocols. Technically, we show the existence of an oracle (namely, PKE-oracle) relative to which key-agreement protocols exist; but it is useless for semi-honest secure realization of symmetric 2-party (deterministic finite) SFE functionalities, i.e. any SFE which can be securely performed relative to this oracle can also be securely performed in the plain model.

Our main result has following consequences.

- There exists an oracle which is useful for some 3-party deterministic SFE; but useless for semi-honest secure realization of any general 2-party (deterministic finite) SFE.
- With respect to semi-honest, standalone or UC security, existence of key-agreement protocols (if used in black-box manner) is only as useful as the commitment-hybrid for general 2-party (deterministic finite) SFE functionalities.

This work advances (and conceptually simplifies) several state-of-the-art techniques in the field of black-box separations:

1. We introduce a general *common-information learning* algorithm (CIL) which extends the "eavesdropper" in prior work [IR89, BM09, HOZ13], to protocols whose message can depend on information gathered by the CIL so far.
2. With the help of this CIL, we show that in a secure 2-party protocol using an idealized PKE oracle, surprisingly, decryption queries are useless.
3. The idealized PKE oracle with its decryption facility removed can be modeled as a collection of *image-testable random-oracles*. We extend the analysis approaches of prior work on random oracle [IR89, BM09, DLMM11, MMP12, HOZ13] to apply to this class of oracles. This shows that these oracles are useless for semi-honest 2-party SFE (as well as for key-agreement).

These information theoretic impossibility results can be naturally extended to yield black-box separation results (cf. [RTV04]).

Keywords: 2-party Deterministic Secure Function Evaluation, Key-agreement Protocols, Public-key Encryption, Black-box Separation, Common Information Learner

[*]University of Virginia, Research done while at Cornell and supported in part by NSF Awards CNS-1217821 and CCF-0746990, AFOSR Award FA9550-10-1-0093, and DARPA and AFRL under contract FA8750-11-2- 0211. mohammad@cs.virginia.edu.

[†]University of California, Los Angeles, Supported by NSF CI Postdoctoral Fellowship. hemanta.maji@gmail.com.

[‡]University of Illinois Urbana-Champaign, Supported in part by NSF grants 1228856 and 0747027. mmp@cs.illinois.edu.

Contents

1 Introduction **1**

1.1 Related Work . . . 2

1.2 Our Contribution . . . 3

1.3 Technical Overview . . . 4

1.4 Overview of the paper . . . 6

2 Oracle Classes **7**

2.1 Image-testable Random Oracle Class . . . 7

2.2 Keyed Version of Image-testable Random Oracle Class . . . 7

2.3 Public-key Encryption Oracle . . . 8

3 Heavy-query Performing Algorithm **8**

4 Modularity of (Keyed Version of) Image-testable Random-oracles **11**

4.1 Notation and Definitions . . . 12

4.2 Imported Result . . . 13

4.3 Some Combinatorial Counting . . . 13

4.4 Local Samplability and Oblivious Re-randomizability . . . 15

5 Common Information Learner for Input-less Protocols **17**

5.1 Notation and Definitions . . . 18

5.2 Independence of Private Views conditioned on Nice-event . . . 19

5.3 Augmentation of Protocols . . . 21

5.4 Reduction in Niceness is Low . . . 21

5.5 Common Information Learner for Input-less Protocols . . . 22

6 Compiling our Decryption Queries **24**

6.1 Complete Proofs . . . 28

7 Common Information Learner for Protocols with Inputs **33**

7.1 Augmentation of Two-party Protocols . . . 34

7.2 Notation and Definitions . . . 34

7.3 Common-Information Learner Properties 34
7.4 Extension of [MMP12] to Image-testable Random-oracles 37

8 Putting Things Together **37**

9 Conclusions and Open Problems **38**

References **41**

A Technical Results **42**
A.1 Result about Independence 42
A.2 Reduction in Niceness 42
A.3 Product Distribution Basics 43

1 Introduction

Public-key encryption (PKE) is an important security primitive in a system involving *more than two parties.* In this work, we ask if PKE could be useful for protecting two mutually distrusting parties against each other, *if there is no other party involved.* More specifically, we ask if the existence of PKE can facilitate 2-party secure function evaluation (SFE). Informally, our main result in this work shows the following:

> *The existence of PKE (as a computational complexity assumption, when used in a black-box manner) is useless for semi-honest secure evaluation of any finite, deterministic 2-party function.*

Here, a complexity assumption being "useless" for a task means that the task can be realized using that assumption alone (in a black-box manner) if and only if it can be realized unconditionally (i.e., information-theoretically).[1] As is typical in this line of research, our focus is on deterministic functions whose domain-size is finite. (However, all our results extend to the case when the domain-size grows *polynomially* in the security parameter; our proofs (as well as the results we build on) do not extend to exponentially growing domain-sizes, though.) Technically, we show an "oracle-separation" result, by presenting a randomized oracle which enables PKE in the information-theoretic setting, but does not enable SFE for any 2-party function for which SFE was impossible without the oracle. Then, using standard techniques, this information theoretic impossibility result is translated into the above black-box separation result [RTV04]. While the above statement refers to semi-honest security, as we shall shortly see, a similar statement holds for security against active corruption, as well.

It is instructive to view our result in the context of "cryptographic complexity" theory [MPR10]: with every (finite, deterministic) multi-party function f, one can associate a computational intractability assumption that there exists a secure computation protocol for f that is secure against semi-honest corruption.[2] Two assumptions are considered distinct unless they can be black-box reduced to each other. Then, the above result implies that secure key agreement (i.e., the interactive analog of PKE) does not belong to the universe of assumptions associated with 2-party functions. However, it is not hard to see that there are 3-party functions f such that a semi-honest secure protocol for f (in the broadcast channel model) is equivalent to a key agreement protocol.[3] Thus we obtain the following important conclusion:

> *The set of computational complexity assumptions associated (in the above sense) with 3-party functions is strictly larger than the set associated with 2-party functions.*

[1]The task here refers to 2-party SFE in the "plain" model. We do not rule out the possibility that PKE is useful for 2-party SFE in a "hybrid" model, where the parties have access to a trusted third party.

[2]This is the simplest form of assumptions associated with functionalities in [MPR10], where a more general framework is presented.

[3]As an example, consider the 3-party function $f(x, y, z) = x \oplus y$. A semi-honest secure protocol π for f over a broadcast channel can be black-box converted to a key-agreement protocol between Alice and Bob, where, say, Alice plays the role of the first party in π with the key as its input, and Bob plays the role of the second and third parties with random inputs. Conversely, a key-agreement protocol can be used as a black-box in a semi-honest secure protocol for f, in which the first party sends its input to the second party encrypted using a key that the two of them generate using the key-agreement protocol.

This answers an open question posed in [MPR10], but raises many more questions. In particular, we ask if the same conclusion holds if we consider $(n+1)$-party functions and n-party functions, for every $n > 2$.

Another consequence of our main result is its implications for SFE secure against *active* corruption. Following a related work in [MMP12], using characterizations of functions that have SFE protocols secure against semi-honest and active corruptions [Kus89, Bea89, MPR09, MPR12], we obtain the following corollary of our main result.

> *The existence of PKE (as a black-box assumption) is exactly as useful as a commitment functionality (given as a trusted third party) for secure evaluation of any finite, deterministic 2-party function. This holds for semi-honest security, standalone active security and UC-security.*

Note that for semi-honest security, the commitment functionality is not useful at all (since semi-honest parties can commit using a trivial protocol), and this agrees with the original statement. The interesting part of the corollary is the statement about active (standalone or UC) security. Commitment is a "minicrypt" functionality that can be implemented using one-way functions (in the standalone setting) or random oracles. PKE, on the other hand, is not a minicrypt primitive [IR89]. Yet, in the context of guaranteeing security for *two* mutually distrusting parties, computing a (finite, deterministic) function, without involving a trusted third party, PKE is no more useful than the commitment functionality.

In the rest of this section, we state our main results more formally, and present an overview of the techniques. But first we briefly mention some of the related work.

1.1 Related Work

Impagliazzo and Rudich [IR89] showed that random oracles are not useful against a computationally unbounded adversary for the task of secure key agreement. This analysis was recently simplified and sharpened in [BM09, HOZ13]. Haitner, Omri, and Zarosim [HOZ12, HOZ13] show that random oracles are essentially useless in any *inputless protocol*.[4]

Following [IR89] many other black-box separation results have appeared (e.g. [Sim98, GMR01, BPR$^+$08, KSY11, MM11]). In particular, Gertner et. al [GKM$^+$00] insightfully asked the question of comparing oblivious-transfer (OT) and key agreement (KA) and showed that OT is strictly more complex (in the sense of [IR89]). Another trend of results has been to prove lower-bounds on the efficiency of the implementation in black-box constructions (e.g. [KST99, GGKT05, LTW05, HHRS07, BM07, BM09, HHRS07]). A complementary approach has been to find black-box reductions when they exist (e.g. [IL89, Ost91, OW93, Hai08, HNO$^+$09]). Also, results in the black-box separation framework of [IR89, RTV04] have immediate consequences for computational complexity theory. Indeed, as mentioned above, separations in this framework can be interpreted as new worlds in Impagliazzo's universe [Imp95].

Our work relies heavily on [MMP12], where a similar result was proven for one-way functions instead of PKE. While we cannot use the result of [MMP12] (which we strictly improve upon) in a

[4]Ideally, a result similar to that of [HOZ13] should be proven in our setting of secure function evaluation too, where parties do have private inputs, as it would extend to randomized functions as well. While quite plausible, such a result remains elusive.

black-box manner, we do manage to exploit the modularity of the proof there and avoid duplicating any significant parts of the proof.

1.2 Our Contribution

For brevity, in the following we shall refer to "2-party deterministic SFE functions with polynomially large domains" simply as SFE functions. Also, we consider security against semi-honest adversaries in the information theoretic setting, unless otherwise specified (as in Corollary 2).

Our main result establishes that there exists an oracle which facilitates key-agreement while being useless to 2-party SFE.

Theorem 1. *There exists an oracle $\mathbb{PKE}$ such that, the following hold:*

- *There is a secure key-agreement protocol (or equivalently, a semi-honest secure 3-party XOR protocol) using $\mathbb{PKE}$.*
- *A general 2-party deterministic function f, with a polynomially large domain, has a semi-honest secure protocol against computationally unbounded adversaries using $\mathbb{PKE}$ if and only if f has a perfectly semi-honest secure protocol in the plain model.*

As discussed below, this proof breaks into two parts — a compiler that shows that the decryption queries can be omitted, and a proof that the oracle without the decryption queries is not useful for SFE. For proving the latter statement, we heavily rely on a recent result from [MMP12] for random oracles; however, this proof is modular, involving a "frontier analytic" argument, which uses a few well-defined properties regarding the oracles. Our contribution in this is to prove these properties for a more sophisticated oracle class (namely, family of image-testable random oracles), rather than random oracles themselves.

As in [MMP12], Theorem 1 translates to a black-box separation of the primitive PKE from non-trivial SFE. Also, it yields the following corollary, that against active corruption, our $\mathbb{PKE}$ oracle is only as useful as the commitment-hybrid model, as far as secure protocols for 2-party SFE is concerned.

Corollary 2. *There exists an oracle $\mathbb{PKE}$ such that, the following hold:*

- *There is a secure key-agreement protocol (or equivalently, a semi-honest secure 3-party XOR protocol) using $\mathbb{PKE}$.*
- *A general 2-party deterministic function f, with a polynomially large domain, has a statistically semi-honest, standalone or UC-secure protocol relative to $\mathbb{PKE}$ if and only if f has a perfectly, resp., semi-honest, standalone or UC-secure protocol in the commitment-hybrid.*

Apart from there results, and their implications to the complexity of 2-party and 3-party functions, we make important technical contributions in this work. As described below, our "common-information learner" is simpler than that in prior work. This also helps us handle a more involved oracle class used to model PKE. Another module in our proof is a compiler that shows that the decryption facility in PKE is not needed in a (semi-honest secure) protocol that uses PKE, even if the PKE is implemented using an idealized oracle.

1.3 Technical Overview

The main result we need to prove (from which our final results follow, using arguments in [MMP12]) is that there is an oracle class $\mathbb{PKE}_\kappa$ relative to which secure public-key encryption (i.e., 2-round key agreement) protocol exists, but there is no secure protocol for any non-trivial SFE function relative to it.

The oracle class $\mathbb{PKE}_\kappa$ is a collection of following correlated oracles:

- $\mathsf{Gen}(\cdot)$: It is a (length-tripling injective) random oracle which maps secret keys sk to respective public keys pk.
- $\mathsf{Enc}(\cdot,\cdot)$: For each public key pk, it is an independently chosen (length tripling injective) random oracle which maps messages m to cipher texts c.
- $\mathsf{Dec}(\cdot,\cdot)$: Given a valid secret key sk and a valid cipher text c it outputs m such that message m was encrypted using public key $pk = \mathsf{Gen}(sk)$.
- Additionally, it provides test oracles Test which output whether a public key pk is a valid public key or not; and whether a cipher text c has been created using a public key pk or not.

Note that without the Test oracle, this oracle class can be used to semi-honest securely perform OT; hence, all 2-party SFE will be trivial relative to it (see discussion in [GKM$^+$00, LOZ12]). The main technical contribution of this paper is the negative result which shows that the above mentioned oracle class $\mathbb{PKE}_\kappa$ is useless for 2-party SFE against semi-honest adversaries.

This is shown in two steps:

1. First, we show that the decryption oracle $\mathsf{Dec}(\cdot,\cdot)$ is not useful against semi-honest adversaries. That is, given a (purported) semi-honest secure protocol ρ for a 2-party SFE f we compile it into another semi-honest secure protocol Π (with identical round complexity, albeit polynomially more query complexity) which has slightly worse security but performs no decryption-queries.
2. Finally, we observe that the oracle class "$\mathbb{PKE}_\kappa$ minus the decryption oracle" is identical to image-testable random-oracles. And we extend the negative result of [MMP12] to claim that this oracle class is useless for 2-party SFE.

The key component in both these steps is the *Common Information Learner* algorithm, relative to image-testable random oracle class. But we begin by introducing image-testable random oracles.

Image-testable Random-oracle Class. It is a pair of correlated oracles (R, T), where R is a (length-tripling injective) random oracle and test oracle T which outputs whether a point in range has a pre-image or not. We consider *keyed-version* of these oracle, where for each key in an exponentially large key-space $\mathbb{K}$ we have an independent copy of image-testable random oracle.

Note that the answer to an image test query can slightly change the distribution of exponentially many other queries; namely, when we know that y is not in the image of R, the answer to any query x for R will not be uniformly distributed (because it cannot be y). However, since the number of tested images are polynomial-size and the number of possible queries to R are exponentially large, this will affect the distribution of the answers by R only negligibly. Also, because of the expansion of the random oracle R, the fraction of the image-size of R is negligibly small relative to the range of R. So an algorithm, with polynomially-bounded query complexity, who queries the test oracle T has negligible chance of getting a positive answer (i.e. an image) without actually calling R. We emphasize that our whole analysis is conditioned on this event (i.e. accidentally discovering y in the image of the oracle) not taking place; and this requires careful accounting of events because it holds only for (polynomially) bounded query algorithms.

Common Information Learner. The common information learner is a procedure that can see the transcript of an oracle-based protocol between Alice and Bob, and by making a polynomial number of publicly computable queries to the oracle, derives sufficient information such that conditioned on this information, the views of Alice and Bob are almost independent of each other. Our common information learner is similar in spirit to those in [IR89, BM09, MMP12, HOZ13] but is different and more general in several ways:

- **Handling Image-Testable Oracles.** Our common information learner applies to the case when the oracle is not just a random oracle, but an image-testable random oracle family.[5]
- **Interaction between Learner and the System.** It is important for the first part of our proof (i.e. compiling out the decryption queries) that the common information learner *interacts with the protocol execution itself.* That is, at each round the information gathered by the common information learner is used by the parties in subsequent rounds. We require the common information learner to still make only a polynomial number or oracle queries while ensuring that conditioned on the information it gathers, the views of the two parties remain almost independent. In showing that the common information learner is still efficient, we show a more general result in terms of an interactive process between an oracle system (the Alice-Bob system, in our case) and a learner, both with access to an arbitrary oracle (possibly correlated with the local random tapes of Alice and Bob).
- **Simpler Description of the Learner.** The common information learner in our work has a simpler description than that in [IR89, BM09, MMP12]. Our learner queries the random oracle with queries that are most likely to be queried by Alice or Bob in a protocol execution. The learner in [BM09, MMP12] is similar, but uses probabilities not based on the actual protocol, but a variant of it; this makes the description of their common information learner more complicated, and somewhat complicates the proofs of the query efficiency of the learner.[6]

Showing that Image-Testable Random Oracles are Useless for SFE. In [MMP12] it was shown that random oracles are useless for SFE. This proof is modular in that there are four specific results that depended on the nature of the oracle and the common information learner. The rest of the proof uses a "frontier analytic" argument that is agnostic to the oracle and the common information learner. Thus, in this work, to extend the result of [MMP12] to a family of image-

[5]The work of [HOZ13] also handles a larger set of oracles than random oracles (called *simple* oracle), but that class is not known to include image-testable oracles as special case [Hai13].

[6][IR89] uses an indirect mechanism to find the heavy queries, and reasoning about their common information learner is significantly more involved.

testable random oracles, we need only ensure that these four properties continue to hold. The four properties are as follows:

1. Alice's message conditioned on the view of the CIL is almost independent of Bob's input. (Lemma 10, item 1.)
2. Safety holds with high probability. (Lemma 10, item 2.)
3. A strong independence property of Alice's and Bob's views conditioned on that of the CIL. (Lemma 10, item 3.)
4. Modularity properties of image-testable random oracles. (Lemma 5 and Lemma 6)

Compiling Out the Decryption Queries. The main idea behind compiling out the decryption queries is that if Alice has created a ciphertext by encrypting a message using a public-key that was created by Bob, and she realizes that there is at least a small (but significant) probability that Bob would be querying the decryption oracle on this ciphertext (since he has the secret key), then she would preemptively send the actual message to him. We need to ensure two competing requirements on the compiler:

1. **Security.** Note that with some probability Alice might send this message even if Bob was not about to query the decryption oracle. To argue that this is secure, we need to argue that a curious Bob *could have* called the decryption oracle at this point, for the same ciphertext.
2. **Completeness.** We need to ensure that in the compiled protocol, Bob will never have to call the decryption oracle, as Alice would have sent him the required decryptions ahead of time.

For security, firstly we need to ensure that Alice chooses the set of encryptions to be revealed based *only* on the common information that Alice and Bob have. This ensures that Bob can sample a view for himself from the same distribution used by Alice to compute somewhat likely decryption queries, and obtain the ciphertext and secret-key from the decryption query made in this view. The one complication that arises here is the possibility that the secret-key in the sampled view is not the same as the secret-key in the actual execution. To rule this out, we rely on the independence of the views of the parties conditioned on the common information. This, combined with the fact that it is unlikely for a valid public-key to be discovered by the system without either party having actually called the key-generation oracle using the corresponding secret-key, we can show that it is unlikely for a sampled view to have a secret-key different from the actual one.

For completeness of the compiler, we again rely on the common information learner to ensure that if Alice uses the distribution based on common information to compute which decryption queries are likely, then it is indeed unlikely for Bob to make a decryption query that is considered unlikely by Alice.

1.4 Overview of the paper

In Section 2 we formally define all relevant oracle classes. We introduce our heavy-querying algorithm in Section 3 for any oracle system. In Section 4 we discuss some of the salient features of the (keyed version of) image-testable random-oracle class. Using the result in Section 3 and one of the properties in Section 4, we show how to construct the common-information learner for oracle systems without any private inputs for parties in Section 5, relative to (keyed version of) image-testable random-oracle class. Based on this CIL, we show in Section 6 that decryption queries are useless in the idealized PKE oracle. Finally, we show that the "PKE minus decryption oracle" is identical to (collection of) image-testable random oracles; and in Section 7 we show how to create a CIL for protocols where parties have private inputs. Leveraging the properties of our oracles and CIL for protocols where parties have private inputs, we show how to obtain our main result Theorem 1 in Section 8.

2 Oracle Classes

In this section we shall introduce the general oracle classes used in our paper. A general class of oracles shall be represented by $\mathbb{O}$. We are interested in three main classes of oracles, each parameterized by the security parameter κ.

2.1 Image-testable Random Oracle Class

The set $\mathbb{O}_\kappa$ consists of the all possible pairs of correlated oracles $O \equiv (R, T)$ of the form:

1. $R : \{0,1\}^\kappa \mapsto \{0,1\}^{3\kappa}$ is an injective function, and
2. $T : \{0,1\}^{3\kappa} \mapsto \{0,1\}$ is defined by: $T(\beta) = 1$ if there exists $\alpha \in \{0,1\}^\kappa$ such that $R(\alpha) = \beta$; otherwise $T(\beta) = 0$.

This class of oracles is known as *image-testable random oracle* class. Based on the length of the query string we can uniquely determine whether it is a query to R or T oracle. Queries to the R oracle and T oracle are, respectively, called R-queries and T-queries. We follow a notational convention. R-queries shall be denoted by α and T-queries shall be represented by β.

2.2 Keyed Version of Image-testable Random Oracle Class

Given a set $\mathbb{K}$ of keys,[7] consider the following oracle $O^{(\mathbb{K})}$: For every $k \in \mathbb{K}$, let $O^{(k)} \in \mathbb{O}_\kappa$. Given a query $\langle k, q \rangle$, where $k \in \mathbb{K}$ and q is the query to an oracle in $\mathbb{O}_\kappa$, answer it with $O^{(k)}(q)$. Let $\mathbb{O}_\kappa^{(\mathbb{K})}$ be the set of all possible oracles $O^{(\mathbb{K})}$. This class of oracle $\mathbb{O}_\kappa^{(\mathbb{K})}$ is called *keyed-version of image-testable random oracle* class. Intuitively, interpret this as a collection of independent copies of oracles from $\mathbb{O}_\kappa$.

[7] Note that the description of the keys in $\mathbb{K}$ is $\mathsf{poly}(\kappa)$; so the size of the set $\mathbb{K}$ could possibly be exponential in κ.

2.3 Public-key Encryption Oracle

We shall use a "PKE-enabling" oracle similar to the one used in [GKM+00]. With access to this oracle, a semantically secure public-key encryption scheme can be readily constructed,[8] yet we shall show that it is useless for SFE. This oracle, which we will call $\mathbb{PKE}_\kappa$ (or simply $\mathbb{PKE}$, when κ is implicit), is a collection of the oracles $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Test}_1, \mathsf{Test}_2, \mathsf{Dec})$ defined as follows:

- Gen: It is a length-tripling injective random oracle from the set of inputs $\{0,1\}^\kappa$ to $\{0,1\}^{3\kappa}$. It takes as input a secret key sk and provides a public-key pk corresponding to it, i.e. $\mathsf{Gen}(sk) = pk$.
- Enc: This is an "encryption" oracle. It can be defined as a collection of length-tripling injective random oracles, keyed by strings in $\{0,1\}^{3\kappa}$. For each key $pk \in \{0,1\}^{3\kappa}$, the oracle implements a random injective function from $\{0,1\}^\kappa$ to $\{0,1\}^{3\kappa}$. When queried with a (possibly invalid) public key pk, and a message $m \in \{0,1\}^\kappa$, this oracle provides the corresponding cipher text $c \in \{0,1\}^{3\kappa}$ for it, i.e. $\mathsf{Enc}(pk, m) = c$.
- Test_1: It is a test function which tests the validity of a public key, i.e. given a public-key pk, it outputs 1 if and only if there exists a secret key sk such that $\mathsf{Gen}(sk) = pk$.
- Test_2: It is a test function which tests the validity of a public key and cipher text pair, i.e. given a public-key pk and cipher text c, it outputs 1 if and only if there exists m such that $\mathsf{Enc}(pk, m) = c$.
- Dec: This is the decryption oracle, from $\{0,1\}^\kappa \times \{0,1\}^{3\kappa}$ to $\{0,1\}^\kappa \cup \{\perp\}$, which takes a secret-key, cipher-text pair (sk, c) and returns (the unique) m such that $\mathsf{Enc}(\mathsf{Gen}(sk), m) = c$. If no such m exists, it outputs $\perp$.

We note that the encryption oracle produces cipher texts for public keys pk irrespective of whether there exists sk satisfying $\mathsf{Gen}(sk) = pk$. This is crucial because we want to key set $\mathbb{K}$ to be defined independent of the Gen oracle.

$\mathbb{PKE}_\kappa$ Without Dec. We note that if we remove the oracle Dec, the above oracle is exactly the same as the image-testable random oracle $\mathbb{O}_\kappa^{(\mathbb{K})}$, with $\mathbb{K} = \{0,1\}^{3\kappa} \cup \{\perp\}$. Here we identify the various queries to $\mathbb{PKE}_\kappa$ with queries to $\mathbb{O}_\kappa^{(\mathbb{K})}$ as follows: $\mathsf{Gen}(sk)$ corresponds to the query $\langle \perp, sk \rangle$, $\mathsf{Enc}(pk, m)$ corresponds to $\langle pk, m \rangle$, $\mathsf{Test}_1(pk)$ corresponds to $\langle \perp, pk \rangle$ and $\mathsf{Test}_2(pk, c)$ corresponds to $\langle pk, c \rangle$.

3 Heavy-query Performing Algorithm

In this section we shall introduce a *Heavy-query Performing* algorithm. Let $\mathbb{O}$ be a finite class of oracles with finite domain D. Our experiment is instantiated by an oracle system Σ and a deterministic "Heavy-query Performer" $\mathcal{H}$ (with implicit parameter σ, see Figure 2).

[8] To encrypt a message of length, say, $\kappa/2$, a random string of length $\kappa/2$ is appended to it, and passed to the "encryption" oracle, along with the public-key.

The oracle system Σ takes a random tape as input which has finite length. Let $\mathbb{S}$ be the set of pairs of random tape r for Σ and oracle $O \in \mathbb{O}$. The system Σ could possibly be computationally unbounded; but its round complexity is finite. Consider the experiment in Figure 1.

1. Let $\mathcal{D}_{\mathbb{S}}$ be a distribution over $\mathbb{S}$ such that $\mathsf{Supp}(\mathcal{D}_{\mathbb{S}}) = \mathbb{S}$. Sample $(r, O) \sim \mathcal{D}_{\mathbb{S}}$.
2. Start an interactive protocol between $\Sigma^O(r)$, i.e. the oracle system Σ with access to oracle O and local random tape r, and the heavy-query performer $\mathcal{H}$.

Figure 1: Protocol between Oracle system Σ and the Heavy-query Performer $\mathcal{H}$.

Heavy-Query-Performer $\mathcal{H}$. After every message sent by the oracle system Σ, perform the following step:

- Repeatedly call Heavy-Query-Finder to obtain a query-answer pair (q^*, a^*); and add the query-answer pair (q^*, a^*) to the transcript T. Until it reports that there are no more heavy queries left.

Heavy-Query-Finder: Let T be the transcript between the oracle system Σ and heavy-query performer $\mathcal{H}$. The messages added by Σ are represented by T_Σ and the set of query-answer pairs added by $\mathcal{H}$ are represented by $T_{\mathcal{H}}$. It has an implicit parameter σ, which is used to ascertain whether a query is heavy or not. Description of the oracle system Σ and the distribution $\mathcal{D}_{\mathbb{S}}$ is also provided to this algorithm.

1. For every $q \in D$, which is not already answered in $T_{\mathcal{H}}$, compute the probability that $\Sigma^{\tilde{O}}(\tilde{r})$ performs the query q when $(\tilde{r}, \tilde{O}) \sim \mathcal{D}_{\mathbb{S}}$ conditioned on transcript T.
2. If there is no query q with probability $\geq \sigma$ then report that there are no more heavy queries left and quit. Otherwise, let q^* be the lexicographically smallest such query.
3. If the answer to q^* is uncertain (given the transcript T) then query O at q^* and obtain the answer a^*. Otherwise, let a^* be the fixed answer to q^*.
4. Return (q^*, a^*).

Figure 2: Heavy-query Performing Algorithm.

We emphasize that the heavy-query performer $\mathcal{H}$ never performs a query unless its answer is uncertain. If the answer to the query q^* in uncertain, we say that the answer to this query has (max) entropy. Let $\mathcal{Q}_\Sigma\left(\langle \Sigma^O(r), \mathcal{H} \rangle\right)$ represent the query-answer set of the oracle system Σ when its local random tape is r, has oracle access to O and is interacting with the heavy-query performer $\mathcal{H}$. Similarly, $\mathcal{Q}_{\mathcal{H}}\left(\langle \Sigma^O(r), \mathcal{H} \rangle\right)$ represents the query-answer set of the heavy-query performer $\mathcal{H}$ which were actually performed to the oracle when Σ has local random tape r and has oracle access to O. Note that $\mathcal{Q}_\Sigma\left(\langle \Sigma^O(r), \mathcal{H} \rangle\right)$ and $\mathcal{Q}_{\mathcal{H}}\left(\langle \Sigma^O(r), \mathcal{H} \rangle\right)$ could possibly be correlated to each other.

Efficiency of the Heavy-query Performer. We argue that the expected query complexity of the heavy-query performer cannot be significantly larger than the query complexity of the system

Σ itself:

Lemma 1 (Efficiency of Heavy-query Performer)**.** *Let $\mathcal{D}_{\mathbb{S}}$ be a joint distribution over the space $\mathbb{S}$ as defined in Figure 1. For every (randomized) oracle system Σ, the expected query complexity of the heavy-query performer $\mathcal{H}$ (presented an Figure 2) is at most $\frac{1}{\sigma}$ times the expected query complexity of the oracle system Σ in the experiment shown in Figure 1. Formally,*

$$\mathop{\mathbb{E}}_{(r,O)\sim\mathcal{D}_{\mathbb{S}}}\left[\left|\mathcal{Q}_{\mathcal{H}}\left(\left\langle\Sigma^{O}(r),\mathcal{H}\right\rangle\right)\right|\right] \leq \frac{\mathbb{E}_{(r,O)\sim\mathcal{D}_{\mathbb{S}}}\left[\left|\mathcal{Q}_{\Sigma}\left(\left\langle\Sigma^{O}(r),\mathcal{H}\right\rangle\right)\right|\right]}{\sigma}$$

In particular, by Markov inequality, the probability that $\mathcal{H}$ asks more than $\frac{\mathbb{E}_{(r,O)\sim\mathcal{D}_{\mathbb{S}}}\left[\left|\mathcal{Q}_{\Sigma}\left(\left\langle\Sigma^{O}(r),\mathcal{H}\right\rangle\right)\right|\right]}{\sigma\cdot\tilde{\sigma}}$ queries is at most $\tilde{\sigma}$.

Before we prove the above mentioned lemma, we mention some highlights of the current proof. The proof is significantly simpler and is more general than the ones presented in [BM09, HOZ13] because our learner is directly working with heavy queries rather than concluding the heaviness of the queries being asked by the learner. Also note that in our setting the oracles might be correlated with local random tape of the system Σ; and the future messages of the oracle system Σ could, possibly, depend on prior messages of $\mathcal{H}$. We also note that the same proof also works in the setting where Σ cannot read the transcript T[9] but $\mathcal{H}$ also considers *queries performed in the future* by Σ while computing the set of heavy-queries.[10]

Proof. The proof proceeds by induction on $|\mathbb{S} = \mathsf{Supp}(\mathcal{D}_{\mathbb{S}})|$.

For the base case, note that $|\mathbb{S}| = 1$ implies $\mathcal{H}$ never queries the oracle; hence, $\left|\mathcal{Q}_{\mathcal{H}}\left(\left\langle\Sigma^{O}(r),\mathcal{H}\right\rangle\right)\right| = 0$ and the induction hypothesis is trivially satisfied.

Assume the hypothesis is true for every $|\mathbb{S}| < s$ and $\mathcal{D}_{\mathbb{S}}$ distribution over $\mathbb{S}$.

Now, for the inductive step, consider $|\mathbb{S}| = s$. Define the following event: "Consider the first place where there is entropy in the message to be added to T by Σ or $\mathcal{H}$ conditioned on T." Note that if there are no entropy in the messages of $\mathcal{H}$, then those queries were never performed to the oracle.

Case 0. If the event mentioned above never takes place then $\left|\mathcal{Q}_{\mathcal{H}}\left(\left\langle\Sigma^{O}(r),\mathcal{H}\right\rangle\right)\right| = 0$ and the hypothesis is trivially satisfied

Case 1. Suppose the first place where the above mentioned event takes place corresponds to a message sent by Σ. Let the possible messages by $m_1, \ldots, m_t$, where $t \geq 2$; and $\mathbb{S}_i \subset \mathbb{S}$ be the set of (r, O) pairs such that Σ and $\mathcal{H}$ interact to generate T and Σ sends m_i as the next message. Define $\mathcal{D}_{\mathbb{S}_i}$ as the distribution $\mathcal{D}_{\mathbb{S}}$ conditioned on generating a sample in $\mathbb{S}_i$; and p_i be the probability that $(r, O) \sim \mathcal{D}_{\mathbb{S}}$ lies in the space $\mathbb{S}_i$.

[9]More specifically, it cannot read $T_{\mathcal{H}}$; note that Σ already knows the part T_{Σ} generated by Σ itself.

[10]Note that if Σ can also read from T then the distribution of future queries is not well defined. But if Σ cannot read T, then future queries are well defined after (r, O) is instantiated.

$$\begin{aligned}
\mathop{\mathbb{E}}_{(r,O)\sim\mathcal{D}_{\mathbb{S}}}\left[\left|\mathcal{Q}_{\mathcal{H}}\left(\left\langle\Sigma^{O}(r),\mathcal{H}\right\rangle\right)\right|\right] &= \sum_{i\in[t]} p_i \cdot \mathop{\mathbb{E}}_{(r,O)\sim\mathcal{D}_{\mathbb{S}_i}}\left[\left|\mathcal{Q}_{\mathcal{H}}\left(\left\langle\Sigma^{O}(r),\mathcal{H}\right\rangle\right)\right|\right] \\
&\leq \sum_{i\in[t]} p_i \cdot \frac{\mathbb{E}_{(r,O)\sim\mathcal{D}_{\mathbb{S}_i}}\left[\left|\mathcal{Q}_{\Sigma}\left(\left\langle\Sigma^{O}(r),\mathcal{H}\right\rangle\right)\right|\right]}{\sigma}, \quad \text{By Induction Hypothesis} \\
&= \frac{\mathbb{E}_{(r,O)\sim\mathcal{D}_{\mathbb{S}}}\left[\left|\mathcal{Q}_{\Sigma}\left(\left\langle\Sigma^{O}(r),\mathcal{H}\right\rangle\right)\right|\right]}{\sigma}
\end{aligned}$$

Case 2. Suppose the first place where the event takes place corresponds to a message sent by $\mathcal{H}$. Let the possible answers to the query q^* be $a_1^*, \dots, a_t^*$, where $t \geq 2$. Define $m_i = (q^*, a_i^*)$. Similar to the previous step we can define $\mathbb{S}_i$, $\mathcal{D}_{\mathbb{S}_i}$ and p_i. Let Σ_i be the system Σ where query q^* is (hardwired) to answer a_i^*, thus the query q^* is never actually performed by the system Σ_i. Further, when $(r, O) \sim \mathcal{D}_{\mathbb{S}_i}$ and $\mathcal{H}$ interacts with Σ_i then $\mathcal{H}$ never performs the query q^* because its answer is fixed to a_i^* in $\mathbb{S}_i$.

$$\begin{aligned}
\mathop{\mathbb{E}}_{(r,O)\sim\mathcal{D}_{\mathbb{S}}}\left[\left|\mathcal{Q}_{\mathcal{H}}\left(\left\langle\Sigma^{O}(r),\mathcal{H}\right\rangle\right)\right|\right] &= 1 + \sum_{i\in[t]} p_i \cdot \mathop{\mathbb{E}}_{(r,O)\sim\mathcal{D}_{\mathbb{S}_i}}\left[\left|\mathcal{Q}_{\mathcal{H}}\left(\left\langle\Sigma_i^{O}(r),\mathcal{H}\right\rangle\right)\right|\right] \\
&\leq 1 + \sum_{i\in[t]} p_i \cdot \frac{\mathbb{E}_{(r,O)\sim\mathcal{D}_{\mathbb{S}_i}}\left[\left|\mathcal{Q}_{\Sigma_i}\left(\left\langle\Sigma_i^{O}(r),\mathcal{H}\right\rangle\right)\right|\right]}{\sigma}, \quad \text{By Induction Hypothesis} \\
&\leq \frac{\mathbb{E}_{(r,O)\sim\mathcal{D}_{\mathbb{S}}}\left[\left|\mathcal{Q}_{\Sigma}\left(\left\langle\Sigma^{O}(r),\mathcal{H}\right\rangle\right)\right|\right]}{\sigma}, \quad \text{By Heaviness of } q^*
\end{aligned}$$

The last inequality is a consequence of the fact that q^* is a heavy query. Intuitively, at least σ fraction of the executions of Σ experience reduction in query complexity by 1 when we define the systems Σ_i. Formally, because q^* is a heavy query we have:

$$\sum_{i\in[t]} p_i \cdot \mathop{\mathbb{E}}_{(r,O)\sim\mathcal{D}_{\mathbb{S}_i}}\left[\left|\mathcal{Q}_{\Sigma_i}\left(\left\langle\Sigma_i^{O}(r),\mathcal{H}\right\rangle\right)\right|\right] \leq \left(\mathop{\mathbb{E}}_{(r,O)\sim\mathcal{D}_{\mathbb{S}}}\left[\left|\mathcal{Q}_{\Sigma}\left(\left\langle\Sigma^{O}(r),\mathcal{H}\right\rangle\right)\right|\right]\right) - \sigma \cdot 1$$

Note that, since Σ has finite round complexity and the domain D of the oracle class is finite, interaction between Σ and $\mathcal{H}$ is finite. Thus, exactly one of the above mentioned cases must occur. This completes the inductive step and, hence, the proof. □

Specific to Image-testable Random-oracles. Relative to the oracle class $\mathbb{O}_{\kappa}^{(\mathbb{K})}$, we can make an assumption that after performing a R-query $\langle k, \alpha\rangle$ and receiving β as answer, $\mathcal{H}$ immediately performs the next query as $\langle k, \beta\rangle$. Note that this query has no entropy (because this query will surely be answered 1); and, hence, need not be actually performed to the oracle.

4 Modularity of (Keyed Version of) Image-testable Random-oracles

In this section we shall prove some properties of (keyed version of) image-testable random oracles. We import a result (see, Equation 1) which says that it is nearly impossible, except for negligible probability, for a polynomially-bounded algorithm to guess y in the image of such an oracle. Further, we show that these oracles have properties (conditioned on suitable events) similar to random oracles: Corollary 4, Lemma 5 and Lemma 6. Corollary 4 is crucial in building CIL for protocols; while Lemma 5 and Lemma 6 are necessary for the analysis of [MMP12] to apply to our oracle class.

4.1 Notation and Definitions

Given a query-answer set P relative to an oracle class $\mathbb{O}$, we represent the set of all oracles $O \in \mathbb{O}$ which are consistent with P as: $\mathbb{O}|_P$, read as $\mathbb{O}$ *restricted* to query-answer set P. A query-answer set P is *consistent* (represented as $\mathsf{Consistent}(P)$) relative to an oracle class $\mathbb{O}$, if $|\mathbb{O}|_P| > 0$. The set of all queries in P is represented by $\mathcal{Q}(P)$.

Definition 1 (Good)**.** *Three query-answer sets P_A, P_B and P_E are called* good, *represented by* $\mathsf{Good}(P_A, P_B, P_E)$, *if* $\mathsf{Consistent}(P_A \cup P_B \cup P_E)$ *and* $P_A \cap P_B \subseteq P_E$.

Specific to oracle classes $\mathbb{O}_\kappa$ and $\mathbb{O}_\kappa^{(\mathbb{K})}$. Given a query-answer set P relative to an oracle in $\mathbb{O}_\kappa$, we denote the number of R-queries in P as $\mathsf{RComp}(P)$. The number of query answer pairs of form $(\beta, 0) \in P$, where β is a T-query is represented by $\mathsf{T_0Comp}(P)$; and number of T-queries of form $(\beta, 1)$ in P is represented by $\mathsf{T_1Guess}(P)$.

We extend these notations to query-answer sets P relative to $\mathbb{O}_\kappa^{(\mathbb{K})}$. The subset of P containing query-answers where the query had key $k \in \mathbb{K}$ is represented by $P|_k$.[11] The definition of RComp, $\mathsf{T_0Comp}$ and $\mathsf{T_1Guess}$ are extended to $P|_k$ and is represented by, respectively, $\mathsf{RComp}(P; k)$, $\mathsf{T_0Comp}(P; k)$ and $\mathsf{T_1Guess}(P; k)$.

Definition 2 (Typical)**.** *A query-answer set P relative to $\mathbb{O}_\kappa^{(\mathbb{K})}$ is* typical, *represented by* $\mathsf{Typical}(P)$, *if* $\mathsf{T_1Guess}(P|_k) = 0$, *for every* $k \in \mathbb{K}$.

Definition 3 (Canonical)**.** *A canonical sequence of query-answer pairs is a sequence of query-answer pairs such that an R-query of form $\langle k, \alpha \rangle$ is immediately followed by a T-query of form $\langle k, \beta \rangle$, where the query $\langle k, \alpha \rangle$ was answered by β.*

In this paper all query-answer sets considered shall be of size $\mathsf{poly}(\kappa)$.

Some Useful Results. In this section we summarize some useful properties of query-answer sets relative to oracle class $\mathbb{O}_\kappa^{(\mathbb{K})}$.

Lemma 2. *Suppose P_A, P_B and P_E are three canonical query-answer sets relative to $\mathbb{O}_\kappa^{(\mathbb{K})}$, such that* $\mathsf{Consistent}(P_A \cup P_B \cup P_E)$. *Then:*

1. $\mathsf{Typical}(P_A \cup P_B \cup P_E)$ *and* $\mathsf{Good}(P_A, P_B, P_E)$ *implies* $\mathsf{Typical}(P_A \setminus P_E)$ *and* $\mathsf{Typical}(P_B \setminus P_E)$.
2. $\mathsf{Typical}(P_A \cup P_B \cup P_E)$ *and* $\mathsf{Good}(P_A, P_B, P_E)$ *implies* $\mathsf{T_1Guess}(P_A \cup P_E) + \mathsf{T_1Guess}(P_B \cup P_E) = \mathsf{T_1Guess}(P_E)$.

[11] We are overloading the notion of "restriction". But the context where a restriction is applied will disambiguate the usage trivially.

Proof. 1. Suppose there exists a query $(\langle k, \beta \rangle, 1) \in P_B \backslash P_E$ but no $\langle k, \alpha \rangle$ is answered with β in the set $P_B \backslash P_E$. But $\mathsf{Typical}(P_A \cup P_B \cup P_E)$ holds. So, there exists α such that $(\langle k, \alpha \rangle, \beta)$ lies in P_A, P_B or P_E. If it lies in P_A, then $(\langle k, \beta \rangle, 1) \in P_A$ (by canonical form); and then $(\langle k, \beta \rangle, 1) \in P_E$ (by good-ness); which is a contradiction. If it lies in P_E, then again $(\langle k, \beta \rangle, 1) \in P_E$ (by canonical form); which is a contradiction. If it lies if P_B, then it must lie in $P_B \cap P_E$; and it reduces to the previous case. So, it is impossible that: $\neg\mathsf{Typical}(P_B \setminus P_E)$. Similarly, we can also show that $\mathsf{Typical}(P_A \setminus P_E)$.

2. Consider a query-answer pair $(\langle k, \beta \rangle, 1) \in P_E$ such that for every $\alpha \in \{0,1\}^\kappa$ we have $(\langle k, \alpha \rangle, \beta) \notin P_E$. Then we claim that: either $(\langle k, \alpha \rangle, \beta) \in P_A \setminus P_E$ or $(\langle k, \alpha \rangle, \beta) \in P_B \setminus P_E$ (but not both). If it is in none of these sets then it will violate $\mathsf{Typical}(P_A \cup P_B \cup P_E)$. If it is in both of these sets then it will violate $\mathsf{Good}(P_A, P_B, P_E)$. Hence we have the result. $\square$

4.2 Imported Result

In this section we present a technical result pertaining to the oracle class $\mathbb{O}_\kappa^{(\mathbb{K})}$. This result intuitively says that an oracle algorithm with polynomially bounded query complexity cannot *guess* an image of $R \xleftarrow{\$} \mathbb{R}_\kappa$ with significant probability, where $\mathbb{R}_\kappa$ is the set of all length-tripling injective oracles with input domain $\{0,1\}^\kappa$.

Lemma 3 (Imported from [KMR89]). *Let $\mathbb{R}_\kappa$ be the set of all injective functions from $\{0,1\}^\kappa$ to $\{0,1\}^{3\kappa}$. Let $\mathcal{A}$ be an oracle algorithm, possibly randomized, with unbounded computational power but restricted to $\mathsf{poly}(\kappa)$ query complexity which has access to an oracle $R \xleftarrow{\$} \mathbb{R}_\kappa$. The probability that $\mathcal{A}^R$ can output β such that:*

1. *There exists α such that $R(\alpha) = \beta$, but*
2. *$\mathcal{A}$ never queried R and received β as answer.*

is at most $\mathsf{negl}(\kappa)$.

Consider any algorithm $\mathcal{A}$ with local random tape $r \xleftarrow{\$} \mathbf{U}$ and oracle access to $O \in \mathbb{O}_\kappa^{(\mathbb{K})}$. Let $P(A^O(r)\text{-}\mathtt{at\text{-}time}\text{-}i)$ be the set of query-answer pairs in $\mathcal{A}$'s view at time i. If the query complexity of $\mathcal{A}$ is $\mathsf{poly}(\kappa)$, then Lemma 3 can be re-stated as follows:

$$\Pr_{\substack{O \xleftarrow{\$} \mathbb{O}_\kappa^{(\mathbb{K})} \\ r \xleftarrow{\$} \mathbf{U}}} \left[\exists i \text{ s.t. } \neg\mathsf{Typical}\left(P(A^O(r)\text{-}\mathtt{at\text{-}time}\text{-}i)\right)\right] = \mathsf{negl}(\kappa) \tag{1}$$

4.3 Some Combinatorial Counting

Let P_A and P_E be canonical sets of query-answer pairs to oracles in $\mathbb{O}_\kappa$ such that $\mathsf{Consistent}(P_A \cup P_E)$ holds. Let $u_E = \mathsf{RComp}(P_E)$, $v_E = \mathsf{T_0Comp}(P_E)$ and $w_E = \mathsf{T_1Guess}(P_E)$. Similarly define $u_{AE} = \mathsf{RComp}(P_A \cup P_E)$, $v_{AE} = \mathsf{T_0Comp}(P_A \cup P_E)$ and $w_{AE} = \mathsf{T_1Guess}(P_A \cup P_E)$.

Lemma 4. *Let P_A and P_E be canonical set of query-answer pairs with respect to oracle class $\mathbb{O}_\kappa$. If $|P_A \cup P_E| = \mathsf{poly}(\kappa)$, then:*

$$\Pr_{O \xleftarrow{\$} \mathbb{O}_\kappa|_{P_E}} [O \in \mathbb{O}_\kappa|_{P_A \cup P_E}] = \frac{N^{3u_E+2w_E}}{N^{3u_{AE}+2w_{AE}}} \times (1 \pm \mathsf{negl}(\kappa)),$$

where $N = 2^\kappa$.

Proof. Note that:

$$|\mathbb{O}_\kappa|_{P_E}| = w_E! \binom{N-w_E}{w_E} \times (N-(u_E+w_E))! \binom{N^3-(u_E+v_E+w_E)}{N-(u_E+w_E)}$$

This is because:

1. u_E elements in $\{0,1\}^\kappa$ are already mapped,
2. w_E elements of $\{0,1\}^{3\kappa}$ receive mappings in $w_E!\binom{N-w_E}{w_E}$ ways, and
3. The number of ways the remaining $N-(u_E+w_E)$ elements of the domain are injectively mapped to the range is $(N-(u_E+w_E))!\binom{N^3-(u_E+v_E+w_E)}{N-(u_E+w_E)}$.

Since $|P_E| = 2u_E + v_E + w_E = \mathsf{negl}(\kappa)N$, we have:

$$\begin{aligned}
n_E = |\mathbb{O}_\kappa|_{P_E}| &= w_E! \binom{N-w_E}{w_E} \times (N-(u_E+w_E))! \binom{N^3-(u_E+v_E+w_E)}{N-(u_E+w_E)} \\
&= (N-u_E)^{w_E} \times (N^3-(u_E+v_E+w_E))^{N-(u_E+w_E)} \times (1 \pm \mathsf{negl}(\kappa)) \\
&= N^{w_E} \times N^{3(N-(u_E+w_E))} \times (1 \pm \mathsf{negl}(\kappa)) \\
&= \frac{N^{3N}}{N^{3u_E+2w_E}} \times (1 \pm \mathsf{negl}(\kappa))
\end{aligned}$$

Similarly, we also have:

$$n_{AE} = \frac{N^{3N}}{N^{3u_{AE}+2w_{AE}}} \times (1 \pm \mathsf{negl}(\kappa))$$

Finally, the lemma follows by observing that:

$$\Pr_{O \xleftarrow{\$} \mathbb{O}_\kappa|_{P_E}} [O \in \mathbb{O}_\kappa|_{P_A \cup P_E}] = \frac{n_{AE}}{n_E} = \frac{N^{3u_E+2w_E}}{N^{3u_{AE}+2w_{AE}}} \times (1 \pm \mathsf{negl}(\kappa)) \quad \square$$

Intuitively, Lemma 4 implies that $\mathsf{T_0Comp}(P_A \cup P_E)$ does not have significant influence. If $\mathsf{T_1Guess}(P_E) = \mathsf{T_1Guess}(P_A \cup P_E) = 0$, then this is nearly equivalent to randomly answering the R-queries in $P_A \setminus P_E$ and the resulting query-answer set being consistent with $P_A \cup P_E$.

The previous result can also be generalized to the oracle class $\mathbb{O}_\kappa^{(\mathbb{K})}$. Let P_A and P_E be sets of query-answer pairs to oracles in $\mathbb{O}_\kappa^{(\mathbb{K})}$. Considering $P_E|_k$, for $k \in \mathbb{K}$, we define $u_{E,k}$, $v_{E,k}$ and $w_{E,k}$ as above. Similarly, by considering $(P_A \cup P_E)|_k$ we define $u_{AE,k}$, $v_{AE,k}$ and $w_{AE,k}$.

Corollary 3. *Let P_A and P_E be canonical set of query-answer pairs with respect to oracle class $\mathbb{O}_\kappa^{(\mathbb{K})}$. If $|P_A \cup P_E| = \mathsf{poly}(\kappa)$, then:*

$$\underset{O \overset{\$}{\leftarrow} \mathbb{O}_\kappa^{(\mathbb{K})}|_{P_E}}{\mathrm{P}} [O \in \mathbb{O}_\kappa^{(\mathbb{K})}|_{P_A \cup P_E}] = \left(\prod_{k \in \mathbb{K}} \frac{N^{3u_{E,k}+2w_{E,k}}}{N^{3u_{AE,k}+2w_{AE,k}}} \right) \times (1 \pm \mathsf{negl}(\kappa))$$

, where $N = 2^\kappa$.

Proof. By taking union bound over keys $k \in \mathbb{K}$ which are queried in $P_A \cup P_E$, we get this result. □

Further, due to Corollary 3, we have:

Corollary 4. *Let P_A, P_B and P_E be canonical set of query-answer pairs with respect to $\mathbb{O}_\kappa^{(\mathbb{K})}$ such that:*

1. *$|P_A \cup P_B \cup P_E| = \mathsf{poly}(\kappa)$,*
2. *$\mathsf{Typical}(P_A \cup P_B \cup P_E)$, and*
3. *$\mathsf{Good}(P_A, P_B, P_E)$.*

Then:

$$\underset{O \overset{\$}{\leftarrow} \mathbb{O}_\kappa|_{P_E}}{\mathrm{P}} [O \in \mathbb{O}_\kappa|_{P_A \cup P_B \cup P_E}] = \frac{N^{2\mathsf{T}_1\mathsf{Guess}(P_E)}}{N^{3\mathsf{RComp}(P_A \setminus P_E)} \times N^{3\mathsf{RComp}(P_B \setminus P_E)}} \times (1 \pm \mathsf{negl}(\kappa))$$

Proof. The result follows directly by using the above corollary with $w_{AE} = \sum_{k \in \mathbb{K}} w_{AE,k} = 0$ and, for the kind of query-answer sets being considered, using the fact that $\mathsf{RComp}(P_A \cup P_B \cup P_E) = \mathsf{RComp}(P_A \setminus P_E) + \mathsf{RComp}(P_B \setminus P_E) + \mathsf{RComp}(P_E)$. □

4.4 Local Samplability and Oblivious Re-randomizability

In this section, we shall prove two properties which we summarize intuitively.

1. Local Samplability: We need Bob to sample hypothetical Bob views without exactly knowing what view Alice has. A crucial step in this is to sample a new query-answer set P_B' which is consistent with P_E; and this sampling has to be independent of the exact query-answer set P_A of Alice.
2. Oblivious Re-randomizability: Once Bob has sampled a hypothetical Bob view, it needs to simulate the view further which includes answering further queries to the (hypothetical) oracle. A crucial step in this is to answer these new queries with answers which are consistent with Alice's query-answer set P_A; and *re-randomize* the part of the oracle which is consistent with the actual Bob query-answer pairs P_B.

Lemma 5 and Lemma 6 intuitively summarize these two properties respectively.

Suppose we are given canonical query-answer sets P_A, P_B, P_E such that $\mathsf{Good}(P_A, P_B, P_E)$ and $\mathsf{Typical}(P_A \cup P_B \cup P_E)$. We are given a canonical query-answer set P_B' such that $\mathsf{Good}(P_A, P_B', P_E)$ and $\mathsf{Typical}(P_A \cup P_B' \cup P_E)$. Note that P_B and P_B' need not be consistent with each other.

Local Samplability. Consider the probability:

$$\begin{aligned}
\frac{\left|\mathbb{O}_\kappa^{(\mathbb{K})}|_{P_A\cup P'_B\cup P_E}\right|}{\left|\mathbb{O}_\kappa^{(\mathbb{K})}|_{P_A\cup P_E}\right|} &= \frac{N^{3\mathsf{RComp}(P_A\cup P_E)+2\mathsf{T}_1\mathsf{Guess}(P_A\cup P_E)}}{N^{3\mathsf{RComp}(P_A\cup P'_B\cup P_E)+2\mathsf{T}_1\mathsf{Guess}(P_A\cup P'_B\cup P_E)}}(1\pm\mathsf{negl}(\kappa)), && \text{by } \textit{Lemma 4}\\
&= \frac{N^{2\mathsf{T}_1\mathsf{Guess}(P_A\cup P_E)}}{N^{3\mathsf{RComp}(P'_B\setminus P_E)}}(1\pm\mathsf{negl}(\kappa)), && \text{by } \mathsf{Good}(P_A\cup P'_B\cup P_E)\\
&= \frac{N^{2\mathsf{T}_1\mathsf{Guess}(P_E)-2\mathsf{T}_1\mathsf{Guess}(P'_B\setminus P_E)}}{N^{3\mathsf{RComp}(P'_B\setminus P_E)}}(1\pm\mathsf{negl}(\kappa)), && \text{by } \textit{Lemma 2}
\end{aligned} \tag{2}$$

This shows that the probability of sampling an oracle in $\mathbb{O}_\kappa^{(\mathbb{K})}|_{P_A\cup P_E}$ which is also in $\mathbb{O}_\kappa^{(\mathbb{K})}|_{P_A\cup P'_B\cup P_E}$ is (nearly) independent of P_A.

Lemma 5 (Local Resampling)**.** *Suppose for polynomial size canonical query-answer set P_A, P'_B and P_E such that $\mathsf{Good}(P_A, P'_B, P_E)$ and $\mathsf{Typical}(P_A\cup P'_B\cup P_E)$ hold. Let p be the probability that $O \xleftarrow{\$} \mathbb{O}_\kappa^{(\mathbb{K})}|_{P_A\cup P_E}$ is also consistent with P'_B.*

Let $\tilde{p} = \frac{N^{2\mathsf{T}_1\mathsf{Guess}(P_E)-2\mathsf{T}_1\mathsf{Guess}(\tilde{P}_B\setminus P_E)}}{N^{3\mathsf{RComp}(\tilde{P}_B\setminus P_E)}}$.

Then we have $p = \tilde{p}(1\pm\mathsf{negl}(\kappa))$.

Proof. The proof is immediate due to Equation 2. □

Oblivious Re-randomizability. Let ΔP be any query-answer set such that $\mathsf{Consistent}(P_A\cup P'_B\cup P_E\cup\Delta P)$. Note that ΔP need not be consistent with P_B and it could have intersection queries with P_A outside P_E. Without loss of generality, we can assume that ΔP does not intersect $P_A\cup P'_B\cup P_E$. We consider the re-randomizing algorithm presented in Figure 3 for Bob. A similar algorithm can also be constructed for Alice and the arguments will be analogous.

A hybrid-argument. Before we analyze the re-randomization algorithm in Figure 3, let us consider an alternate *hybrid* re-randomization algorithm (in Figure 4) which is easier to analyze.

We shall condition on the following event: "The mapping is injective."

Define $U = P_A\cup P'_B\cup P_E\cup\Delta P$.

Note that this event occurs with probability $1-\mathsf{negl}(\kappa)$, because $|U| = \mathsf{poly}(\kappa)$. Conditioned on this event, the following two are identical:

1. Sample an oracle $O \xleftarrow{\$} \mathbb{O}_\kappa^{(\mathbb{K})}|_{P_A\cup P'_B\cup P_E}$ Compute the probability that O is consistent with ΔP.

2. Sample an oracle $O \xleftarrow{\$} \mathbb{O}_\kappa^{(\mathbb{K})}|_{P_A\cup P_B\cup P_E}$. Relative to this oracle run the hybrid-experiment of Figure 4. Compute the probability that queries in $\mathcal{Q}(\Delta P)$ receive identical output as in ΔP.

To complete the argument, note that $\sum_{k\in\mathbb{K}} |E_{k,\mathsf{global}}| \leq |U| = \mathsf{poly}(\kappa)$. Thus, the difference in the experiments Figure 4 and Figure 3 is $\mathsf{negl}(\kappa)$.

Suppose Alice has private query-answer sequence P_A, Eve has P_E and Bob has P_B. Assume that Bob has been provided with P'_B; and $\mathsf{Typical}(P_A \cup P \cup P_E)$ and $\mathsf{Good}(P_A, P, P_E)$ hold, for $P \in \{P_B, P'_B\}$.
Let D be the set of R-queries in P_B which are not already included in $\mathcal{Q}(P_E \cup P'_B)$. We re-emphasize that the queries in $\mathcal{Q}(P_B) \cap \mathcal{Q}(P'_B)$ outside $\mathcal{Q}(P_E)$ could possible by inconsistently answered.
Initialize a global set $R_{\mathsf{local}} = \emptyset$.
$\mathsf{Query\text{-}Answering}$ (q)

1. If q is answered in $P_E \cup P'_B$ use that answer.
2. If $q = \langle k, \alpha \rangle$ is a new R-query and $q \in D$, answer with $a \xleftarrow{\$} \{0,1\}^{3\kappa}$. Add $\langle k, a \rangle$ to R_{local}.
3. If q is a T-query which has already in R_{local} then answer 1.
4. Otherwise (i.e. if the conditions above are not met) forward the query to the actual oracle and obtain the answer a.

Figure 3: Bob's algorithm to answer future queries using re-randomization (oblivious to P_A).

Lemma 6 (Oblivious Re-randomizing). *Let P_A, P_B, P'_B and P_E be polynomial size canonical query-answer sets such that:*

1. $\mathsf{Good}(P_A, P_B, P_E)$ *and* $\mathsf{Typical}(P_A \cup P_B \cup P_E)$*, and*
2. $\mathsf{Good}(P_A, P'_B, P_E)$ *and* $\mathsf{Typical}(P_A \cup P'_B \cup P_E)$.

Let ΔP be a polynomial size query-answer set such that $\mathsf{Consistent}(P_A \cup P'_B \cup P_E \cup \Delta P)$*. Define p as the probability that $O \xleftarrow{\$} \mathbb{O}_\kappa^{(\mathbb{K})}|_{P_A \cup P'_B \cup P_E}$ is consistent with ΔP.*

Sample $O \xleftarrow{\$} \mathbb{O}_\kappa^{(\mathbb{K})}|_{P_A \cup P_B \cup P_E}$ and run the oblivious re-randomizing algorithm provided in Figure 3. This defines a distribution over polynomial size query answer sets. Let $\tilde{p}$ be the probability of ΔP under this distribution.

Then: $p = \tilde{p}(1 \pm \mathsf{negl}(\kappa))$.

5 Common Information Learner for Input-less Protocols

In this section, we show how to construct the *Common-information Learner* for 2-party protocols which have no inputs. We need to construct an eavesdropper algorithm which publicly captures all the common information between Alice and Bob private views. First, in Lemma 7, we show that conditioned on suitable events joint-views of Alice and Bob is (close to) a product distribution. Next, we show that starting with Alice-Bob joint views which are already close to product distribution, the updated joint-views (after one of them sends a message in a round) cannot be significantly far from product views (see Lemma 8). Finally, we prove our main result (Lemma 9) that constructs a curious eavesdropper algorithm such that with high probability Alice-Bob joint views are close to product distribution based on the public-view at the end of each round of the protocol.

Initialize a global set $R_{\mathsf{local}} = \emptyset$ and $E_{k,\mathsf{global}} = \emptyset$ (for each $k \in \mathbb{K}$).

1. If q is answered in $P_E \cup P'_B$ use that answer.
2. If $q = \langle k, \alpha \rangle$ is a new R-query and $q \in D$, answer with $a \xleftarrow{\$} \{0,1\}^{3\kappa} \setminus E_{\mathsf{global}}$. Add $\langle k, a \rangle$ to R_{local}.
3. If q is a T-query which has already in R_{local} then answer 1.
4. Otherwise (i.e. if the conditions above are not met) forward the query to the actual oracle and obtain the answer a. If the $q = \langle k, \beta \rangle$ is a T-query and $a = 0$ then add β to $E_{k,\mathsf{global}}$.

Figure 4: Hybrid-experiment.

5.1 Notation and Definitions

In this section we shall consider three party protocols between Alice, Bob and Eve relative to oracle class $\mathbb{O}_\kappa^{(\mathbb{K})}$ over broadcast channel. We shall consider only a restriction of Eve, namely *Public Query Strategies.* They have the following properties:

1. Eve has no private state. Her view is completely public.
2. In every round i after Alice or Bob has added a message m_i to the public transcript, Eve is invoked.
3. When invoked, Eve performs a list of queries to the oracle $O \in \mathbb{O}_\kappa^{(\mathbb{K})}$ and broadcasts the sequence of query-answer pairs $P_{E,i}$.

Thus Eve's view consists of the public transcript m generated by Alice and Bob; and the sequence of query-answer pairs P_E she broadcasts to all parties. We can write Eve's view $V_E = (m, P_E)$.

Alice's view also includes her private random tape r_A and her private query-answers P_A. So, we can write $V_A = (r_A, m, P_A, P_E)$. Similarly, we can also write $V_B = (r_B, m, P_B, P_E)$.

Note that public-query strategy Eve is deterministic and her view is a subset of each party's view. Our sample space is joint distribution of Alice-Bob views when $r_A \xleftarrow{\$} \mathbf{U}$, $r_B \xleftarrow{\$} \mathbf{U}$ and $O \xleftarrow{\$} \mathbb{O}_\kappa^{(\mathbb{K})}$. We shall use bold-math notation to represent the random variables, for example $\mathbf{P}_A$ represents the random variable for Alice private query-answer set and $\mathrm{P}[\mathbf{P}_A = P_A]$ is the probability that P_A is Alice's private random query-answer set when Alice-Bob joint views is sampled according to the sample space mentioned above.

Definition 4 (Normal Protocol Form)**.** *A three party protocol between Alice, Bob and (public query strategy) Eve is in normal form, if:*

1. *In every round Alice or Bob sends a message; followed by a sequence of query-answer pairs from Eve.*
2. *In rounds $i = 1, 3, \ldots$ Alice sends the message m_i; and in $i = 2, 4, \ldots$ Bob sends a message.*

3. *In every round i after Alice/Bob has sent the message m_i, Eve broadcasts $P_{E,i}$.*

4. *Alice, Bob and Eve always perform canonical queries.*

A triplet of views V_A, V_B and V_E are called possible, represented as $\mathsf{Possible}(V_A, V_B, V_E)$, if: There exists an oracle O such that when the three party protocol is run with Alice local randomness r_A and Bob local randomness r_B then the views produced for Alice, Bob and Ever are respectively P_A, P_B and P_E. Note that $\mathsf{Possible}(V_A, V_B, V_E)$ implies $\mathsf{Consistent}(P_A, P_B, P_E)$.

Definition 5 (Nice Views)**.** *Alice, Bob and Eve views in a normal protocol are called* nice*, represented as* $\mathsf{Nice}(V_A, V_B, V_E)$ *if:* $\mathsf{Possible}(V_A, V_B, V_E)$*,* $\mathsf{Good}(P_A, P_B, P_E)$ *and* $\mathsf{Typical}(P_A \cup P_B \cup P_E)$*.*

Views at end of rounds. Suppose the set of messages sent by Alice-Bob is $m = m_1 m_2 \ldots$. We represent $m^{(i)} = m_1 \ldots m_i$, i.e. the message at the end of round i. Similarly, we define $P_E^{(i)} = P_{E,1} \ldots P_{E,i}$, i.e. the query-answer pairs added by Eve till the end of round i. Analogously, we also define $V_A^{(i)} = (r_A, m^{(i)}, P_A^{(i)}, P_E^{(i)})$ and $V_B^{(i)} = (r_B, m^{(i)}, P_B^{(i)}, P_E^{(i)})$.

We shall use $\mathbf{V}_E^{(i)}$ as the distribution over Eve views at the end of round i after V_E is drawn from the underlying sample space of complete Alice-Bob views.

Augmented Protocols. Suppose we are provided with a two-party protocol π between Alice and Bob. An augmentation of π, represented as π^+, with a third party Eve is a three-party protocol over a broadcast channel. In every round of the augmented protocol, Alice/Bob broadcasts a message followed by Eve. In this paper, we shall only consider augmentations of two-party protocols with public-query strategies.

Let us clarify the notion of augmented protocols with some examples:

1. Suppose Alice and Bob have next message functions as defined by π. This next message function could be such that it ignores any message not sent by Alice or Bob. In this case, the augmentation of π by a third party has no effect on the behavior of Alice/Bob. The third party simply adds additional messages to the public transcript. For example, consider the augmentation of key-agreement protocol with eavesdropper strategies (as in Corollary 5).

2. On the other hand, the next message functions for Alice and Bob as defined in π could behave differently in 2-party and 3-party setting. If Alice sees additional messages in the public transcript (i.e. messages which were not sent by Alice or Bob) then it could run a different next message function; otherwise it runs the 2-party next message function.

3. In particular, we shall deal with augmented protocols where Alice/Bob's 3-party version of next message protocol will use the messages sent by the third party Eve to form a wrapper around the original 2-party next message generation function. For example, consider the compiler presented in Section 6.

5.2 Independence of Private Views conditioned on Nice-event

In this section we shall prove the following result.

$\mathsf{Simulate}(V_E)$:
Comment: Essentially, this is a rejection sampling algorithm which outputs "nice" views.

1. Let $A(V_E) = \{V_A : \exists\ V_B \text{ s.t. } \mathsf{Possible}(V_A, V_B, V_E)\}$ and, similarly, $B(V_E) = \{V_B : \exists\ V_A \text{ s.t. } \mathsf{Possible}(V_A, V_B, V_E)\}$.
2. Let $\mathcal{A}(V_E)$ be the distribution over $A(V_E)$ which puts weight on V_A proportional to $\frac{1}{N^{\mathsf{3RComp}(P_A \setminus P_E)}}$, where $N = 2^\kappa$. And $\mathcal{B}(V_E)$ be the distribution over $B(V_E)$ which puts weight on V_B proportional to $\frac{1}{N^{\mathsf{3RComp}(P_B \setminus P_E)}}$.
3. Sample $V_A \sim \mathcal{A}(V_E)$, $V_B \sim \mathcal{B}(V_E)$. If $\mathsf{Nice}(V_A, V_B, V_E)$ then output (V_A, V_B); otherwise output $\mathsf{Simulate}(V_E)$.

Figure 5: Independent Simulation of Alice and Bob private Views.

Lemma 7. *Consider the sampling algorithm in Figure 5. Let V_A, V_B and V_E be Alice, Bob and Eve (partial) views, respectively, in an oracle protocol π with access to $O \in \mathbb{O}_\kappa^{(\mathbb{K})}$. If V_A, V_B, V_E are such that $\mathsf{Nice}(V_A, V_B, V_E)$ holds then:*

$$\frac{\mathrm{P}_{(\mathbf{V}_A,\mathbf{V}_B)\sim\mathsf{Simulate}(V_E)}[\mathbf{V}_A = V_A, \mathbf{V}_B = V_B]}{\mathrm{P}[\mathbf{V}_A = V_A, \mathbf{V}_B = V_B | \mathbf{V}_E = V_E, \mathsf{Nice}(\mathbf{V}_A, \mathbf{V}_B, \mathbf{V}_E)]} = (1 \pm \mathsf{negl}(\kappa))$$

Intuitively, this result states that, conditioned on "nice" event occurring, the distribution of Alice-Bob joint views is close to a product distribution also conditioned on the same "nice" event.

Proof. For nice V_A, V_B and V_E, we are interested in computing the following probability:

$$\mathrm{P}[\mathbf{V}_A = V_A, \mathbf{V}_B = V_B | \mathbf{V}_E = V_E, \mathsf{Typical}(\mathbf{P}_A \cup \mathbf{P}_B \cup \mathbf{P}_E), \mathsf{Good}(\mathbf{P}_A, \mathbf{P}_B, \mathbf{P}_E)]$$

Consider the following manipulation:

$$\begin{aligned}
&\mathrm{P}[\mathbf{V}_A = V_A, \mathbf{V}_B = V_B | \mathbf{V}_E = V_E, \mathsf{Typical}(\mathbf{P}_A \cup \mathbf{P}_B \cup \mathbf{P}_E), \mathsf{Good}(\mathbf{P}_A, \mathbf{P}_B, \mathbf{P}_E)] \\
&= \frac{\mathrm{P}[\mathbf{V}_A = V_A, \mathbf{V}_B = V_B, \mathsf{Typical}(\mathbf{P}_A \cup \mathbf{P}_B \cup \mathbf{P}_E), \mathsf{Good}(\mathbf{P}_A, \mathbf{P}_B, \mathbf{P}_E), \mathbf{V}_E = V_E]}{\mathrm{P}[\mathsf{Typical}(\mathbf{P}_A \cup \mathbf{P}_B \cup \mathbf{P}_E), \mathsf{Good}(\mathbf{P}_A, \mathbf{P}_B, \mathbf{P}_E), \mathbf{V}_E = V_E]}, \\
&\qquad \text{By Bayes' Rule} \\
&= \frac{\mathrm{P}[\mathbf{V}_A = V_A, \mathbf{V}_B = V_B, \mathbf{V}_E = V_E]}{\mathrm{P}[\mathsf{Typical}(\mathbf{P}_A \cup \mathbf{P}_B \cup \mathbf{P}_E), \mathsf{Good}(\mathbf{P}_A, \mathbf{P}_B, \mathbf{P}_E), \mathbf{V}_E = V_E]}, \\
&\qquad \text{Because } \mathsf{Nice}(V_A, V_B, V_E) \text{ holds} \\
&= \frac{\mathrm{P}[\mathbf{r}_A = r_A, \mathbf{r}_B = r_B, \mathbf{m} = m, \mathbf{P}_A = P_A, \mathbf{P}_B = P_B, \mathbf{P}_E = P_E]}{\mathrm{P}[\mathsf{Typical}(\mathbf{P}_A \cup \mathbf{P}_B \cup \mathbf{P}_E), \mathsf{Good}(\mathbf{P}_A, \mathbf{P}_B, \mathbf{P}_E), \mathbf{V}_E = V_E]}, \\
&\qquad \text{Expanding the random variables } \mathbf{V}_A, \mathbf{V}_B, \mathbf{V}_E \\
&= \frac{\mathrm{P}[\mathbf{r}_A = r_A, \mathbf{r}_B = r_B, \mathbf{P}_A = P_A, \mathbf{P}_B = P_B, \mathbf{P}_E = P_E]}{\mathrm{P}[\mathsf{Typical}(\mathbf{P}_A \cup \mathbf{P}_B \cup \mathbf{P}_E), \mathsf{Good}(\mathbf{P}_A, \mathbf{P}_B, \mathbf{P}_E), \mathbf{V}_E = V_E]}, \\
&\qquad \text{Because } \mathbf{m} \text{ is a deterministic function of } \mathbf{r}_A, \mathbf{r}_B, \mathbf{P}_A, \mathbf{P}_B \\
&= \frac{\mathrm{P}[\mathbf{r}_A = r_A]\,\mathrm{P}[\mathbf{r}_B = r_B]\,\mathrm{P}[\mathbf{P}_E = P_E] \times \mathrm{P}[\mathbf{P}_A = P_A, \mathbf{P}_B = P_B | \mathbf{P}_E = P_E]}{\mathrm{P}[\mathsf{Typical}(\mathbf{P}_A \cup \mathbf{P}_B \cup \mathbf{P}_E), \mathsf{Good}(\mathbf{P}_A, \mathbf{P}_B, \mathbf{P}_E), \mathbf{V}_E = V_E]},
\end{aligned}$$

$$\begin{aligned}
&\text{By Chain Rule and Independence of } \mathbf{r}_A, \mathbf{r}_B \\
&= c_{V_E} \times \frac{N^{2\mathsf{T}_1\mathsf{Guess}(P_E)}}{N^{3\mathsf{RComp}(P_A \setminus P_E)} \times N^{3\mathsf{RComp}(P_B \setminus P_E)}} \times (1 \pm \mathsf{negl}(\kappa)), \\
&\text{where } c_{V_E} \text{ is a constant dependent only on } V_E \text{ and using Corollary 4} \\
&= d_{V_E} \times \frac{1}{N^{3\mathsf{RComp}(P_A \setminus P_E)}} \times \frac{1}{N^{3\mathsf{RComp}(P_B \setminus P_E)}} \times (1 \pm \mathsf{negl}(\kappa)), \\
&\text{where } d_{V_E} \text{ is a constant dependent only on } V_E
\end{aligned}$$

The result directly proves Lemma 7 (See Equation 5 in Appendix A.1). □

5.3 Augmentation of Protocols

Given a two party protocol π we augment it with a third party protocol Eve_π as described in Figure 6. The algorithm is parametrized by a parameter σ which shall be chosen sufficiently small.

1. Interpret the two-party oracle protocol π as the oracle system Σ in Figure 1. Messages produced by Alice or Bob in round i is interpreted as the message of Σ.
2. Define $\mathbb{O} = \mathbb{O}_\kappa^{(\mathbb{K})}$ and $\mathcal{D}_\mathbb{S}$ as the uniform distribution over $\mathbb{O}$ and the space of local random tapes of Alice and Bob.
3. Let Eve_π be the heavy-query performer algorithm in Figure 2 instantiated with a suitably small parameter σ.

Figure 6: Augmentation of an input-less two-party protocol with a public-query strategy Eve_π.

5.4 Reduction in Niceness is Low

We shall prove the following result:

Lemma 8. *Let π be a normal two party protocol between Alice and Bob, where parties have access to an oracle from the class $\mathbb{O}_\kappa^{(\mathbb{K})}$. Consider an augmentation of this two-party protocol using a public-query strategy Eve_π described in Figure 6. Then, for every round i, we have:*

$$\mathrm{P}\left[\left(\mathsf{Nice}^{(1)} \wedge \cdots \wedge \mathsf{Nice}^{(i)}\right) \wedge \left(\neg\mathsf{Long}^{(i)}\right)\right] \geq 1 - \Theta(\psi^2 \sigma i)$$

Intuitively, this lemma implies that views of the parties in a random execution of protocol π along with the Eve's view (w.h.p.) remains "nice"; and Eve performs at most $2\psi/\sigma^2$ queries.

Proof. Let $\mathbb{A}^{(0)} = \mathbb{B}^{(0)} = \emptyset$ and $\mathbb{C}^{(0)}$ is the set containing the empty transcript. Recursively define $\mathbb{A}^{(i+1)}$, $\mathbb{B}^{(i+1)}$ and $\mathbb{C}^{(i+1)}$ as follows:

1. $\mathbb{A}^{(i+1)} = \left\{V_E^{(i+1)} : \mathsf{Long}^{(i+1)} \wedge \exists V_E^{(i)} \in \mathbb{C}^{(i)} \text{ s.t. } V_E^{(i+1)} \models V_E^{(i)}\right\}$. Here, $V_E^{(i+1)} \models V_E^{(i)}$ is true if and only if Eve's view $V_E^{(i+1)}$ was $V_E^{(i)}$ at the end of round i.

2. $\mathbb{B}^{(i+1)} = \left\{ V_E^{(i+1)} : \neg\mathsf{Long}^{(i+1)} \ \wedge \ \mathrm{P}[\mathsf{Nice}^{(i+1)} | V_E^{(i+1)}] < \frac{1}{2} \ \wedge \ \exists V_E^{(i)} \in \mathbb{C}^{(i)} \text{ s.t. } V_E^{(i+1)} \models V_E^{(i)} \right\}$, and
3. $\mathbb{C}^{(i+1)} = \left\{ V_E^{(i+1)} : V_E^{(i+1)} \notin \mathbb{A}^{(i+1)} \cup \mathbb{B}^{(i+1)} \ \wedge \ \exists V_E^{(i)} \in \mathbb{C}^{(i)} \text{ s.t. } V_E^{(i+1)} \models V_E^{(i)} \right\}$.

Intuitively we are doing the following: Consider a complete execution. Suppose i is the smallest index such that $V_E^{(i)}$ for this execution $\mathsf{Long}^{(i)}$ or $\mathrm{P}[\mathsf{Nice}(\mathbf{V}_A^{(i)}, \mathbf{V}_B^{(i)}, V_E^{(i)}) | V_E^{(i)}] < \frac{1}{2}$ holds. If $\mathsf{Long}^{(i)}$ holds then we account for this execution in the set $\mathbb{A}^{(i)}$. Otherwise if $\mathrm{P}[\mathsf{Nice}(\mathbf{V}_A^{(i)}, \mathbf{V}_B^{(i)}, V_E^{(i)}) | V_E^{(i)}] < \frac{1}{2}$ holds then we account for this execution in the set $\mathbb{B}^{(i)}$.

We define:

1. $a_i = \mathrm{P}[V_E^{(i)} \in \mathbb{A}^{(i)}]$,
2. $b_i = \mathrm{P}[\neg\mathsf{Nice}^{(i)} \ \wedge \ V_E^{(i)} \in \mathbb{B}^{(i)}]$, and
3. $c_i = \mathrm{P}[\neg\mathsf{Nice}^{(i)} \ \wedge \ V_E^{(i)} \in \mathbb{C}^{(i)}]$.

We have the following properties:

1. Note that $a_0 = b_0 = c_0 = 0$. Because, initially there are no queries in Alice, Bob and Eve views.
2. $\mathrm{P}[\mathbb{A}^{(i)} \cup \mathbb{B}^{(i)}] \leq a_i + 2b_i$, by averaging arguments.
3. We have: $\sum_{i \in [n]} a_i \leq \sigma$ (Lemma 1).
4. For $i < n$, we have $b_{i+1} + c_{i+1} \leq c_i + \Theta(\psi^2 \sigma) + \mathsf{negl}(\kappa)$ (Lemma 14). Further, this recurrence has the following telescoping property: $c_k + \sum_{i \in [k]} b_i \leq \Theta(\psi^2 \sigma i) + \mathsf{negl}(\kappa)$, for $\sigma(\kappa) = 1/\mathsf{poly}(\kappa)$.
5. $\mathrm{P}[V_E^{(i)} \notin \mathbb{C}^{(i)}] \leq (a_1 + \cdots + a_i) + 2(b_1 + \cdots + b_i) \leq \sigma + 2(\Theta(\psi^2 \sigma i) + \mathsf{negl}(\kappa)) = \Theta(\psi^2 \sigma i) + \mathsf{negl}(\kappa)$.
6. $\mathrm{P}[V_E^{(i)} \in \mathbb{C}^{(i)} \ \wedge \ \mathsf{Nice}^{(i)}] = 1 - (a_1 + \cdots + a_i) - 2(b_1 + \cdots + b_i) - c_i \geq 1 - \Theta(\psi^2 \sigma i) - \mathsf{negl}(\kappa)$.

Thus, we have

$$\mathrm{P}[\mathsf{Nice}^{(1)} \ \wedge \ \cdots \ \wedge \ \mathsf{Nice}^{(i)}] \geq 1 - \Theta(\psi^2 \sigma i), \text{ and}$$
$$\mathrm{P}[\neg\mathsf{Long}^{(i)}] \geq 1 - \sigma,$$

for every $i \leq n$. □

5.5 Common Information Learner for Input-less Protocols

In this section we prove the following result:

Lemma 9 (Common Information Learner in Input-less Protocols)**.** *Let π be an input-less randomized two-party protocol in normal form using a random oracle $O \in \mathbb{O}_\kappa^{(\mathbb{K})}$ with round complexity n and query complexity (at most) ψ, for some key set $\mathbb{K}$. Let Eve_π be an eavesdropper strategy as defined in Figure 6 (for a suitable choice of the parameter $\sigma = 1/\mathsf{poly}(\kappa)$). For every $\varepsilon = 1/\mathsf{poly}(\kappa)$, there exists $\sigma = 1/\mathsf{poly}(\kappa)$ such that: Let $V_E^{(i)}$ be the restriction of $V_E \sim \mathbf{V}_E$ to the end of round i. With probability $1-\varepsilon$ over the choice of V_E, the following holds:*

1. $(1-\varepsilon)$**-Independence:** *For every round i, the following distributions are ε-close:*
$$\left((\mathbf{V}_A^{(i)}|V_E^{(i)}) \times (\mathbf{V}_B^{(i)}|V_E^{(i)})\right) \text{ and } \left((\mathbf{V}_A^{(i)}, \mathbf{V}_B^{(i)})|V_E^{(i)}\right)$$

2. ε**-Lightness:** *For every round i and query $q \notin \mathcal{Q}(P_E^{(i)})$ it holds that:*
$$\Pr_{V_A^{(i)} \sim (\mathbf{V}_A^{(i)}|V_E^{(i)})}[q \in \mathcal{Q}(P_A^{(i)})] \le \varepsilon \text{ and } \Pr_{V_B^{(i)} \sim (\mathbf{V}_B^{(i)}|V_E^{(i)})}[q \in \mathcal{Q}(P_B^{(i)})] \le \varepsilon$$

3. **Strong** $(1-\varepsilon)$**-Independence:** *For every round i, if Alice sends the $i+1$'st message, the following distributions are ε-close:*
$$\left((\mathbf{V}_A^{(i+1)}|V_E^{(i)}) \times (\mathbf{V}_B^{(i)}|V_E^{(i)})\right) \text{ and } \left((\mathbf{V}_A^{(i+1)}, \mathbf{V}_B^{(i)})|V_E^{(i)}\right)$$

Proof. First Property. Using Lemma 8 we can conclude the following: There exists a polynomial p^* such that,
$$\Pr[\neg\mathsf{Nice}^{(1)} \ \vee \ \cdots \ \vee \ \neg\mathsf{Nice}^{(n)} \ \vee \ \mathsf{Long}^{(i)}] \le p^*(\sigma, n, \psi)$$

By averaging argument (Markov Inequality), one can conclude that, over the choice of V_E, with probability at least $1 - \sqrt{p^*(\sigma, n, \psi)}$, we shall have:
$$\Pr[\neg\mathsf{Nice}^{(1)} \ \vee \ \cdots \ \vee \ \neg\mathsf{Nice}^{(n)} \ \vee \ \mathsf{Long}^{(n)}|V_E] \le \sqrt{p^*(\sigma, n, \psi)}$$

For such a V_E, note that $\neg\mathsf{Long}^{(n)}(V_E)$ must hold. Then we have that: $\Pr[\neg\mathsf{Nice}^{(1)} \vee \cdots \vee \neg\mathsf{Nice}^{(n)}|V_E] \le \sqrt{p^*(\sigma, n, \psi)}$.

Let $V_E^{(i)}$ be a truncation of V_E up to the end of any round i. Then it holds that:
$$\Pr[\mathsf{Nice}^{(i)}|V_E^{(i)}] \ge 1 - \sqrt{p^*(\sigma, n, \psi)}$$

Note that conditioned on $\mathsf{Nice}^{(i)}$, the joint distribution of Alice-Bob views $(\mathbf{V}_A^{(i)}, \mathbf{V}_B^{(i)}|V_E^{(i)}, \mathsf{Nice}^{(i)})$ is close to a product distribution (Lemma 7). This implies that (by Lemma 15):
$$\mathbf{\Delta}\left((\mathbf{V}_A^{(i)}, \mathbf{V}_B^{(i)}|V_E^{(i)}), (\mathbf{V}_A^{(i)}|V_E^{(i)}) \times (\mathbf{V}_B^{(i)}|V_E^{(i)})\right) \le (\mathsf{poly}(\psi) \cdot p^*(\sigma, n, \psi))^{\Theta(1)}$$

By setting $\sigma = \mathsf{poly}(\varepsilon, 1/n, 1/\psi, 1/\kappa)$ for a sufficiently large polynomial, we get the first property of the result.

Second Property. Consider a complete view of Eve V_E as above. Since $\neg\mathsf{Long}(V_E)$ holds, this implies that the eavesdropper did not find any more heavy-queries. Thus, we know that the probability of any query $q \notin P_E$ is at most σ when Alice-Bob joint views are sampled according to the distribution $(\mathbf{V}_A^{(i)}, \mathbf{V}_B^{(i)} | V_E^{(i)})$. In particular, the probability that q is in $\mathcal{Q}(P_A^{(i)})$ is at most σ when sampled according to the distribution $(\mathbf{V}_A^{(i)} | V_E^{(i)})$.

By using sufficiently small σ, we can ensure that the second condition holds.

Third Property. The third property follows from the first and second properties and a simple change of parameter over ε (and using the value of ε that makes all properties hold). A very similar argument was used in in [MMP12]. Here we sketch the argument for sake of completeness.

Suppose we have obtained a V_E conditioned on which, the joint distribution of views of Alice and Bob till the end of round i is close to a product distribution. We claim that the same holds (with a weaker parameter), even if we look at the distribution of Alice till generating the next message m_{i+1}. The reason is the following. Since the views of Alice and Bob are ε-close, if we sample the views of Alice and Bob independently, and then execute Alice to get m_{i+1} consistently with Bob (as if we are running the real experiment in which Alice and Bob are sampled jointly) and call this the imaginary experiment, the distribution of the parties in the imaginary experiment remains ε close to generating m_{i+1} from the real execution. Now we use the second property over the imaginary experiment: till generating m_{i+1}, Alice will ask any of Bob's private queries (i.e. those not already in V_E) only with probability $O(\varepsilon \cdot \psi)$. The reason is that, as long a Alice has not asked Bob's private queries, her execution continues independently of Bob; also, using a union bound and the second property, the probability of hitting and of Bob's queries is at most $\psi \cdot \varepsilon$. $\square$

As a direct consequence of Lemma 9 we have the following corollary:

Corollary 5. *There does not exist a secure key-agreement protocol relative to oracle class $\mathbb{O}_\kappa^{(\mathbb{K})}$, for any key set $\mathbb{K}$.*

6 Compiling our Decryption Queries

In this section we show that a family of PKE-enabling oracles is only as useful as a family of image testable random oracles, for semi-honest SFE. Combined with the result that this image testable random oracle family is useless for SFE, we derive the main result in this paper, that PKE is useless for semi-honest SFE.

As pointed out by [GKM$^+$00], care must be taken in modeling such an oracle so that it does not allow oblivious transfer. In our case, we need to separate it from not just oblivious transfer but any non-trivial SFE.

In our proof we shall use the oracle $\mathbb{PKE}_\kappa$ defined in Section 2.3. This oracle facilitates public-key encryption (by padding messages with say $\kappa/2$ random bits before calling Enc), and hence key agreement. But, as mentioned before, by omitting the Dec oracle, the collection $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Test}_1, \mathsf{Test}_2)$ becomes an image-testable random oracle family $\mathbb{O}^{(\mathbb{K})}$. As we will see in Section 7.4, an image-testable random oracle is not useful for SFE or key agreement. The challenge is to show that even given the decryption oracle Dec, which does help with key-agreement, the oracle remains useless for SFE. [GKM$^+$00] addressed this question for the special case of oblivious transfer, relying on properties that are absent from weaker (yet non-trivial) SFE functionalities. Our approach is to instead show that the decryption facility is completely useless in SFE, by giving a carefully compiled protocol whereby the parties help each other in finding decryptions of ciphertexts without accessing Dec oracle, while retaining the security against honest-but-curious adversaries. We show the following.

Theorem 6. *Suppose Π is an N-round 2-party protocol with input domain $\mathcal{X} \times \mathcal{Y}$, that uses the oracle $\mathbb{PKE}_\kappa$. Then for any polynomial* poly*, there is an N-round protocol Π^* using the oracle $\mathbb{O}_\kappa^{(\mathbb{K})}$ that is as secure as Π against semi-honest adversaries, up to a security error of $|\mathcal{X}||\mathcal{Y}|/\mathsf{poly}(\kappa)$.*

Below we present the compiler used to prove this theorem, and sketch why it works. The full proof appears in Section 6.1.

The Idea Behind the Compiler. Conceptually the compiler is simple: each party keeps track of the ciphertexts that it *created* that the other party becomes "capable of" decrypting and sends the message in the ciphertext across at the right time. This will avoid the need for calling the decryption oracle. But we need to also argue that the compilation preserves security: if the original protocol was a secure protocol for some functionality, then so is the compiled protocol. To ensure this, a party, say Bob, should reveal the message in an encryption it created only if there is a high probability that Alice (or a curious adversary with access to Alice) *can* obtain that message by decryption. Further, the fact that Bob found out that Alice could decrypt a ciphertext should not compromise Bob's security. This requires that just based on common information between the two parties it should be possible to accurately determine which ciphertexts each party can possibly decrypt. This is complicated by the fact that the protocol can have idiosyncratic ways of transferring ciphertexts and public and private keys between the parties, and even if a party *could* carry out a decryption, it may choose to not extract the ciphertext or private key implicit in its view. By using the common information learner for image testable random oracles, it becomes possible to

Definition of the Compiler. Given a 2-party protocol Π, with input domains $\mathcal{X} \times \mathcal{Y}$, we define the compiled protocol Π^* below. Π has access to $\mathbb{PKE}_\kappa$, where as Π^* will have access to the interface of $\mathbb{PKE}_\kappa$ consisting only of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Test}_1, \mathsf{Test}_2)$ (or equivalently, to $\mathbb{O}_\kappa^{(\mathbb{K})}$ as described in Section 2). For convenience, we require a normal form for Π that before making a decryption query $\mathsf{Dec}(sk, c)$ a party should make queries $\mathsf{Gen}(sk), \mathsf{Test}_1(pk)$ and $\mathsf{Test}_2(pk, c)$ where pk was what was returned by $\mathsf{Gen}(sk)$.

We define Π^* in terms of a 3-party protocol involving $\mathsf{Alice}_0, \mathsf{Bob}_0, \mathsf{Eve}$, over a broadcast channel. In the following we will define Alice_0 and Bob_0; this then defines an (inputless) system Σ which consists of them interacting with each other internally, while interacting with an external party; in Σ, the inputs to Alice_0 and Bob_0 are picked uniformly at random. Then, Eve is defined to be $\mathcal{H}$ for the system Σ, as defined in Figure 6: after each message from Σ (i.e., from Alice or Bob), Eve responds with a set of publicly computable queries to the oracle. Finally, Π^* is defined as follows: Alice runs Alice_0 and Eve internally, and Bob runs Bob_0 and Eve.[12]

So to complete the description of the compiled protocol, it remains to define the programs Alice_0 and Bob_0. We will define Alice_0; Bob_0 is defined symmetrically.

[12]Note that Eve follows a deterministic public-query strategy, and can be run by both parties. Alternately, in Π^*, one party alone could have run Eve. But letting both parties run Eve will allow us to preserve the number of rounds exactly, when consecutive messages from the same party are combined into a single message.

. . . .

Procedure 1. Alice_0 internally maintains the state of an execution of Alice's program in Π (denoted by Alice_Π). In addition, Alice_0 maintains a list L_A of entries of the form (m, pk, c), one for every call $\mathsf{Enc}(pk, m) = c$ that Alice_Π has made so far, along with such triples from the (secondary) messages from Bob_0.

Corresponding to a single message m_i from Alice in Π, Alice_0 will send out two messages — a primary message m_i and a secondary message $C_{A,i}$ (with an intermediate message from Eve) — as follows. (For the sake of brevity we ignore the boundary cases $i = 1$ and $i-1$ being the last message in the protocol; they are handled in a natural way.)

The list $L_{A,}$, before receiving the $i - 1^{\text{st}}$ message, is denoted by $L_{A,i-2}$.

- On receiving m_{i-1} and $C_{B,i-1}$ from Bob_0 (and the corresponding messages from Eve), first Alice sets $L_{A,i-1} := L_{A,i-2} \cup C_{B,i-1}$ (where $C_{B,i-1}$ is parsed as a set of entries of the form (m, pk, c)).
- Then Alice_0 passes on m_{i-1} to Alice_Π, and Alice_Π is executed. During this execution Alice_Π is given direct access to $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Test1}, \mathsf{Test2})$; but for every query of the form $\mathsf{Dec}(sk, c)$ from Alice_Π, Alice_0 obtains $pk = \mathsf{Enc}(sk)$ and checks if any entry in $L_{A,i-1}$ is of the form (m, pk, c) for some m. If it is, Alice_0 will respond to this query with m. Otherwise Alice_0 responds with $\perp$. At the end of this computation, the message output by Alice_Π is sent out as m_i.

  Also Alice updates the list $L_{A,i-1}$ (which was defined above as $L_{A,i-2} \cup C_{B,i-1}$) to $L_{A,i}$ by including in it a tuple (m, pk, c) for each encryption query $\mathsf{Enc}(pk, m) = c$ that Alice_Π made during the above execution.
- Next it reads a message from Eve. Let $T^{(i)}$ denote the entire transcript at this point (including messages sent by Alice_0, Bob_0 and Eve). Based on this transcript Alice_0 computes a set $D_{\mathrm{B}}^{T^{(i)}}$ of ciphertexts that Bob is "highly likely" to be able to decrypt in the next round, but has not encrypted itself,[13] and then creates a message $C_{A,i}$ that would help Bob decrypt all of them without querying the decryption oracle. The algorithm Assist_A used for this is detailed below in Procedure 2. Alice finishes her turn by sending out $C_{A,i}$.

. . . .

Procedure 2. Procedure Assist_A for computing $C_{A,i}$:

For each possible view V_B of Bob_Π at the point $T^{(i)}$ is generated let,

$$\begin{aligned} d_{\mathrm{B}}(V_B) := \{(pk, c) | \exists sk \text{ s.t. } & [\mathsf{Gen}(sk) = pk], [\mathsf{Test}_1(pk) = 1], [\mathsf{Test}_2(pk, c) = 1] \in V_B \\ \text{and } & \nexists m \text{ s.t. } [\mathsf{Enc}(pk, m) = c] \in V_B\}. \end{aligned}$$

We define the set

$$D_{\mathrm{B}}^{T^{(i)}} := \{(pk, c) | \, \mathrm{P}[(pk, c) \in d_{\mathrm{B}}(V_B) | T^{(i)}] > \delta\} \tag{3}$$

[13]The threshold δ used in defining $D_{\mathrm{B}}^{T^{(i)}}$ by itself does not make it *highly likely* for the honest Bob to be able to decrypt a ciphertext. However, as we shall see, this will be sufficient for a *curious* Bob to be able to decrypt with high probability.

where the probability is over a view V_B for Bob_Π sampled conditioned on $T^{(i)}$, in the interaction between Σ (i.e., Alice_0 and Bob_0 with a random input pair) and Eve.[14] The threshold δ which will be set to an appropriately small quantity (larger than, but polynomially related to, σ associated with Eve).

The message $C_{A,i}$ is a set computed as follows: for each $(pk, c) \in D_{\mathrm{B}}^{T^{(i)}}$, if there is an m such that $\mathsf{Enc}(pk, m) = c$ appears in $L_{A,i}$, then the triple (m, pk, c) is added to $C_{A,i}$. If for any (pk, c), if there is no such m, then the entire protocol is aborted.

. . . .

Security of the Compiled Protocol. To formally argue the security of the compiled protocol we must show an honest-but-curious *simulator* with access to either party in an execution of Π, which can simulate the view of an honest-but-curious adversary in Π^*. Here we do allow a small (polynomially related to σ), but possibly non-negligible simulation error. We give a detailed analysis of such a simulation in Section 6.1. Below we sketch some of the important arguments.

Firstly, it must be the case that the probability of Alice_0 aborting in Π^* while computing a secondary message $C_{A,i}$, is small. Suppose, with probability p Alice fails to find an encryption for some $(pk, c) \in D_{\mathrm{B}}^{T^{(i)}}$. Then, by the independence property guaranteed by Lemma 9, with about probability δp this Alice execution takes place in conjunction with a Bob view V_B such that $(pk, c) \in d_{\mathrm{B}}(V_B)$. This would mean that with close to probability δp we get an execution of the original protocol Π in which (pk, c) is present in the parties' views, but neither Alice nor Bob created this ciphertext. This probability must then be negligible.

The more interesting part is to show that it is safe to reveal an encrypted message, when there is only a small (but inverse polynomial) probability that the other party would have decrypted it. For concreteness, consider when the honest-but-curious adversary has access to Alice. In the execution of Π^* it sees the messages $C_{B,i}$ that are sent by Bob (assuming Bob does not abort). These contain the messages for each (pk, c) pair in D_{A}^{T} where T is the common information so far. So we need to show that the simulator would be able to extract all these messages as well. Consider a $(pk, c) \in D_{\mathrm{A}}^{T}$. If Alice's view contains an Enc query that generates c, or a Gen query that generates pk, then the simulator can use this to extract the encrypted message. Otherwise it samples a view A' for Alice consistent with T, but conditioned on $(pk, c) \in d_{\mathrm{A}}(A')$ (such A' must exist since $(pk, c) \in D_{\mathrm{A}}^{T}$). Then A' does contain a secret key sk' for pk that Alice will use to decrypt the ciphertext.

However, note that the view of the oracle in A' need not be consistent with the given oracle. Thus it may not appear meaningful to use sk' as a secret key. But intuitively, if it is the case that with significant probability Alice did not generate pk herself, then it must have been generated by Bob, and then the only way Alice could have carried out the decryption is by extracting Bob's secret key from the common information. Thus this secret key is fixed by the common information. Further, by sampling an Alice view in which a secret key for pk occurs, this secret key must, with high probability agree with the unique secret key implicit in the common information. Formalizing this intuition heavily relies on the independence characterization: otherwise the common information need not fix the secret key, even if it fixes the public key.

[14]Even though we define Alice_0 in terms of a probability that is in terms of the behavior of a system involving Alice_0, we point out that this probability is well-defined. This is because the probability computed in this round refers only to the behavior of the system up till this round. Also Eve, which is also part of the system generating $T^{(i)}$, depends only on the prior messages from Alice_0.

In Section 6.1 we give a detailed proof of security of Π^*, by defining a complete simulation, and using a coupled execution to analyze how good the simulation is. We show that the compiled protocol is as secure as the original protocol up to a security error of $O|\mathcal{X}||\mathcal{Y}|(N(\sigma/\delta+\delta)) = O(1/\mathsf{poly})$ by choosing appropriate parameters, assuming $|\mathcal{X}||\mathcal{Y}|$ is polynomial.

The proof relies on Lemma 9. It shows that even when the protocol allows the parties to use the information from the common information learner, it holds that the views of the two parties (in an inputless version of the protocol considered in the proof) are nearly independent of each other's, conditioned on the common information gathered by Eve.

6.1 Complete Proofs

In this section we prove Theorem 6. In Figure 7a we illustrate how Π^* works.

Firstly, we note that Π^* is efficient if Π is.

Claim 1. *Suppose Π is a protocol using the $\mathbb{PKE}_\kappa$ oracle in which Alice and Bob make at most $\mathsf{poly}(\kappa)$ oracle queries and communicates at most $\mathsf{poly}(\kappa)$ bits. Then Π^* is a protocol using the $\mathbb{O}_\kappa^{(\mathbb{K})}$ oracle, with at most $\mathsf{poly}(\kappa, 1/\delta)$ number of oracle queries and communication.*

This follows from the efficiency of Eve (Lemma 1), and the fact that for any T, we have $|D_{\mathrm{A}}^T| \leq \mathsf{poly}(\kappa, 1/\delta)$ and (similarly $|D_{\mathrm{B}}^T|$), since $|d_{\mathrm{B}} V_B| \leq \mathsf{poly}(\kappa)$ for all possible V_B.

Below we argue that Π^* is as secure as Π. For honest-but-curious corruption, it is enough to consider the case when exactly one of the two parties is corrupt (this subsumes the correctness requirement of when both parties are honest). W.l.o.g. we consider when the adversary has access to Alice, and Bob is honest. For any such adversary Adv in the compiled protocol, we define a corresponding adversary Sim in the original protocol, as shown in Figure 7b. The simulator is defined as follows:

. . . .

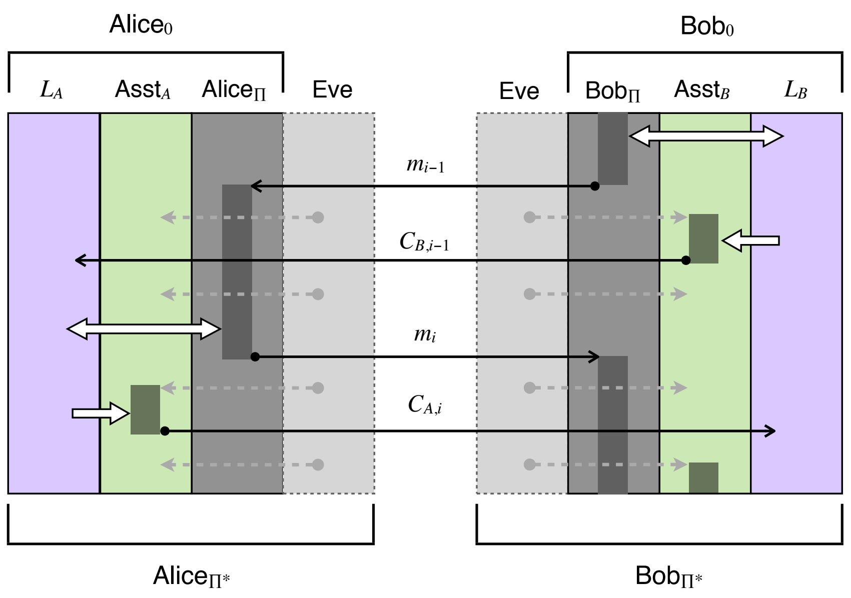
Procedure 3. A semi-honest adversary Sim controlling Alice_Π in an execution of Π (with access to the oracle $\mathbb{PKE}$.)

Goal: to simulate the view of an arbitrary (poly-query) adversary Adv controlling Alice_{Π^*} in an execution of Π^*.

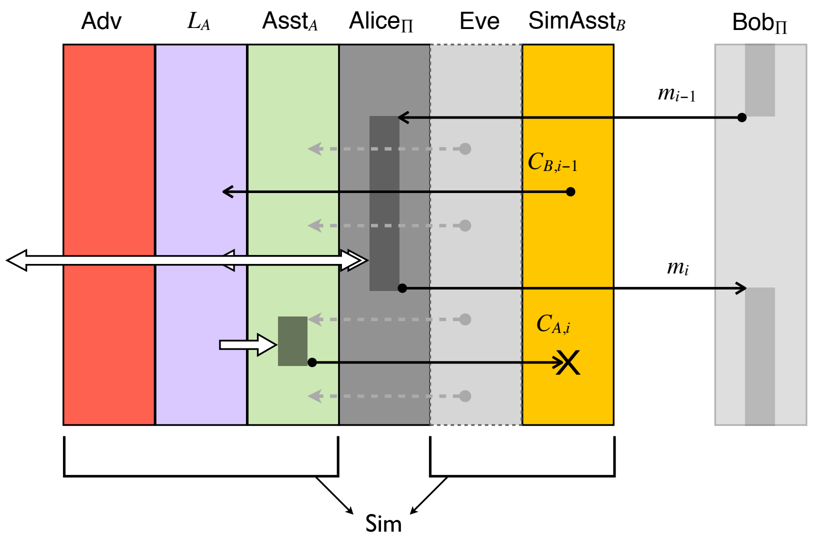
Sim runs Adv, as well as the honest compiled protocol Alice_{Π^*}, with the following modifications:

- Dec queries made by the copy of Alice_Π within Alice_{Π^*} are answered by the actual $\mathbb{PKE}$ oracle (rather than by looking up L_A); but L_A is still kept updated as in Π^*.
- Even if Assist_A (within Alice_{Π^*}) aborts, the copy of Alice_Π is not aborted. (But the rest of Alice_{Π^*} and Adv would be terminated.)
- The secondary messages C_B from Bob_0 are simulated using a procedure $\mathsf{SimAssist}_B$, described below. (The primary messages m are received from Bob_Π.) If $\mathsf{SimAssist}_B$ aborts, then the execution of Alice_Π alone is continued, and the rest of Alice_{Π^*} and Adv are terminated.

. . . .



(a) The compiled protocol between Alice_{Π^*} and Bob_{Π^*}. It shows how Alice_0 produces m_i and $C_{A,i}$ on receiving m_{i-1} and $C_{B,i-1}$ from Bob_0 and the interleaving messaged from Eve (shown by dotted lines). m_i is produced by Alice_Π, but with its Dec queries answered from the list L_A. $C_{A,i}$ is produced by Assist_A, again by looking up L_A. The darkened bars within Alice_Π and Assist_A indicate the execution between when the inputs for these algorithms become available and when they produce their outputs. L_A is updated with the message $C_{B,i-1}$ as well as the Enc queries made by Alice_Π during its computation.



(b) Simulator when Adv corrupts Alice_{Π^*}. The simulator runs the $\mathsf{SimAssist}_B$ algorithm to simulate the messages from Assist_B. Also, it modifies the execution of Alice_Π so that the Dec queries to the oracle are answered by the oracle itself (but L_A is still kept updated), and even if Assist_A aborts, Alice_Π does not. This ensures that Sim is an hones-but-curious adversary for Π.

Figure 7: The compiled protocol Π^* and the simulation to prove its security

Procedure 4. Procedure $\mathsf{SimAssist}_B$:

Input: a view V_A for Alice_0, when it expects a message C_B from Bob_0.

Output: a simulated message C_B.

Let T denote the transcript in the view V_A (which includes the messages from Eve). Let D^T_{A} be as defined in Equation 3 (but defined from the point of view of Bob, instead of Alice). The goal of this algorithm is to generate a set C_B such that for each $(pk, c) \in D^T_{\mathrm{A}}$, there is a (unique) entry (m, pk, c) in T. (This algorithm makes at most $|D^T_{\mathrm{A}}|$ queries (all decryptions) to the oracle.)

First compute the set D^T_{A} (based on the transcript T). Then for each $(pk, c) \in D^T_{\mathrm{A}}$ try to add an (m, pk, c) to the set C_B as follows:

(i) If for some m there is a query of the form $[\mathsf{Enc}(pk, m) = c]$ in V_A, add (m, pk, c) to C_B.

(ii) Else, if for some sk there is a query of the form $[\mathsf{Gen}(sk) = pk]$ in V_A, then make the oracle query $\mathsf{Dec}(sk, c)$ and if the oracle returns m, add (m, pk, c) to C_B.

(iii) Else, the algorithm proceeds as follows: sample a view V'_A for Alice_0 in the execution of Π^* with uniformly random inputs, but conditioned on transcript being T, and conditioned on $(pk, c) \in d_{\mathrm{A}}(V'_A)$ (this is possible since $(pk, c) \in D^T_{\mathrm{A}}$). Since $(pk, c) \in d_{\mathrm{A}}(V'_A)$ there exists a query $[\mathsf{Gen}(sk') = pk]$ in V'_A. Query the decryption oracle with $\mathsf{Dec}(sk', c)$. If the oracle answers with $\perp$ then the algorithm aborts. Else, if the answer is m, then it adds (m, pk, c) to C_B.

. . . .

Note that in the execution of the original protocol Π with Sim controlling Alice_Π, Sim behaves as an honest-but-curious adversary; it lets the execution of Π proceed unhindered, in particular, giving Alice_Π direct access to the $\mathbb{PKE}$ oracle (including Dec) and never aborting its execution.

To argue that this gives a good simulation (up to a small error) of the execution of Π^* with Adv controlling Alice_{Π^*}, we define a *coupled execution* of the two protocols (see Figure 8). For this coupled execution, we sample a single oracle (including Dec, though the compiled protocol and Adv do not have access to this part), a single pair of inputs (x, y) for Alice and Bob (where x and y are independently chosen uniformly at random), a single pair of random tapes for Alice_Π and Bob_Π, and a single random tape for Adv. Then we choose a random tape for the rest of the simulator.[15] These deterministically define an execution for the entire coupled system. The coupled execution terminates the first time any of the failure events (Extraction failure, Lookup failure or Abort failure) occurs.

It is easy to verify the following claim:

Claim 2. *In an instance of the coupled execution defined in Figure 8, if none of the failure events occur, then the final view of* Adv *in the two executions are identical.*

This relies on the fact that if $\mathsf{SimAssist}_B$ or Assist_A finds an m such that $\mathsf{Enc}(pk, m) = c$, and if $\mathsf{Gen}(sk) = pk$, then it must be the case that $\mathsf{Dec}(sk, c) = m$.

[15]The simulator, as described above, uses extra randomness in $\mathsf{SimAssist}_B$. The randomness is used essentially to sample a secret key (by sampling a view V'_A) when it is not found in Alice's view. It is easy to derandomize this by having it pick the most probable secret key instead. But for the sake of a direct analysis, we allow this randomness.

Π^* execution	Π execution
An honest-but-curious adversary Adv runs along with Alice_{Π^*} (which consists of Alice_{Π}, L_A, Assist_A and Eve, with the Dec queries of Alice_{Π} being asnwered using L_A).	Sim runs along with Alice_{Π}; Sim consists of Adv, L_A, Assist_A, and Eve. All Alice_{Π} queries to the oracle are answered by the oracle.
Adv (in both executions) has received m_{i-1} from Bob_{Π}, as well as the following message from Eve, and is expecting a message $C_{B,i-1}$ from Assist_B next.	
Adv receives the message $C_{B,i-1}$ from Assist_B implemented by Bob_{Π^*} (and L_A and Eve's view are updated using it).	Sim uses $\mathsf{SimAssist}_B$ to compute $C_{B,i-1}$. If $\mathsf{SimAssist}_B$ does not abort, then Sim passes $C_{B,i-1}$ to Adv (and L_A and Eve's view are updated using it).
Extraction failure: In the Π execution $\mathsf{SimAssist}_B$ aborts (because it fails to extract the message m for some $(pk, c) \in D_{\mathrm{A}}^{T^{(i)}}$).	
Alice_{Π^*} passes m_{i-1} to Alice_{Π}, and Alice_{Π} is executed to compute m_i. Dec queries made by Alice_{Π} are answered by looking up $L_{A,i-1}$.	Alice_{Π} receives m_{i-1} and computes m_i. Dec queries are answered by directly asking the oracle (ignoring the answer from $L_{A,i-1}$).
Lookup failure: In the Π^* execution, the answer that is computed for a decryption query by Alice_{Π}, by looking up the list $L_{A,i-1}$ is different from what Dec oracle would give.	
Alice_{Π^*} computes $T^{(i)}$ (by running Eve) and then invokes Assist_A to compute $C_{A,i}$, which is forwarded to Bob. Assist_A (and Alice_{Π^*}) aborts during this computation if for some $(pk, c) \in D_{\mathrm{B}}^{T^{(i)}}$, it cannot find an entry of the form $\mathsf{Enc}(pk, m) = c$ is in $L_{A,i}$.	Eve and Assist_A are run as in Π^*. $C_{A,i}$ is not forwarded to Bob.
Abort failure: In Π^* execution Alice_{Π^*} (i.e., Assist_A) aborts.	

Figure 8: **The coupled Π^* and Π executions.**

Now we prove several claims about Π^* and the coupled execution that imply that probability of any failure event occuring is small, no matter what the inputs (x, y) are, or equivalently when the inputs are chosen uniformly at random (since the size of the domains $|\mathcal{X} \times \mathcal{Y}|$ is polynomial). All the proofs use Lemma 9 applied to the protocol Π^* with uniform inputs. To facilitate this, we define two distributions, $\mathbf{V}_{AB}^{(i)}$ and $\mathbf{V}_{A\times B}^{(i)}$ as follows. Fix the uniform distribution over the inputs (x, y) to Alice and Bob. $\mathbf{V}_{AB}^{(i)}$ is simply the joint distribution over the views of Alice_0 and Bob_0 just before Assist_A is invoked to compute $C_{A,i}$. Let $T^{(i)}$ denote the transcript at that point. $\mathbf{V}_{A\times B}^{(i)}$ can be defined as follows. Draw $T^{(i)}$ according to its marginal distribution in $\mathbf{V}_{AB}^{(i)}$, and then *independently* sample a view for Alice_0 and a view for Bob_0, according to their marginal distributions in $\mathbf{V}_{AB}^{(i)}$, conditioned on the value of $T^{(i)}$ that was sampled.

For notational convenience we shall write $\mathbf{V}_{AB}^{(T)}$ and $\mathbf{V}_{A\times B}^{(T)}$ to denote the distributions $\mathbf{V}_{AB}^{(i)}$ and

$\mathbf{V}^{(i)}_{A\times B}$ conditioned on $T^{(i)} = T$ (where i is implicit in T).

Claim 3. *For any $\varepsilon > 1/\mathsf{poly}(\kappa)$, there is a choice of $\sigma > 1/\mathsf{poly}(\kappa)$ such that, for all i,*

$$\Delta(\mathbf{V}^{(i)}_{AB}, \mathbf{V}^{(i)}_{A\times B}) \leq \varepsilon. \tag{4}$$

This claim follows by a direct application of Lemma 9 noting that the oracle used by Π^* is indeed $\mathbb{O}^{(\mathbb{K})}_{\kappa}$.

Claim 4. *Suppose Equation 4 holds. Then, the probability of Alice (or Bob) aborting in Π^* due to Assist_A (rep. Assist_B) failing is $O(N\varepsilon/\delta)$, where N is an upperbound on the number of messages exchanged in Π.*

Proof. Consider the probability of Assist_A aborting in round i of Π^*. This happens when Assist_A fails to find a query of the form $[\mathsf{Enc}(pk, m) = c]$ in her list of oracle queries for some $(pk, c) \in D^T_{\mathrm{B}}$, where T is the common information gathered till that point. Suppose this happens with probability p_T, conditioned on T. Note also that $(pk, c) \in D^T_{\mathrm{B}}$ implies that with probability at least δ, there is no oracle query of the form $[\mathsf{Enc}(pk, m) = c]$ in Bob's view (but $[\mathsf{Test}_1(pk) = 1]$ and $[\mathsf{Test}_2(pk, c) = 1]$ are). Then, in the distribution $\mathbf{V}^{(T)}_{A\times B}$, with probability at least δp_T, neither the view of Alice_Π nor the view of Bob_Π contains a query of the form $[\mathsf{Enc}(pk, m) = c]$ for some $(pk, c) \in D^T_{\mathrm{B}}$, but $[\mathsf{Test}_1(pk) = 1]$ and $[\mathsf{Test}_2(pk, c) = 1]$ are in the view of Bob_Π. We call this event an anomaly.

Let p_i be the probability that Alice aborts while computing the i^{th} message in Π^*. p_i is the expected value of p_T when T is distributed over transcripts $T^{(i)}$. From the above, the probability of an anomaly in round i is at least $p_i\delta$, when the views are sampled according to $\mathbf{V}^{(i)}_{A\times B}$. Then, by Lemma 9, with probability at least $p_i\delta - \varepsilon$, an anomaly occurs after i messages in an execution of Π^*. But this anomaly can happen with only negligible probability. Hence p_i is at most negligibly larger than ε/δ. Then, by the union bound, probability of Assist_A (or Assist_B) aborting at any round in Π^* is $O(N\varepsilon/\delta)$ where N is the total number of messages exchanged in Π. □

Claim 5. *In the coupled execution of Π^* and Π, the probability that a Lookup failure occurs is $O(N\delta)$).*

Proof. A Lookup failure occurs while Alice_{Π^*} computes m_i if a Dec query (pk, c) made by Alice_Π is not listed in $L_{A,i-1}$ (and is hence answered by $\perp$), but in fact, Dec oracle would have given a valid message m. Now, for a Lookup failure to occur, an Abort failure must not have occurred before it (since the coupled execution terminates the first time a failure event occurs). So we can assume that all queries in $D^{T^{(i)}}_{\mathrm{A}}$ were actually answered in $C_{B,i-1} \subseteq L_{A,i-1}$. This means that (pk, c) is not in $D^{T^{(i)}}_{\mathrm{A}}$ and neither is it in the list of local Enc queries Alice made. That is, the view of Alice_Π V_A must be such that $(pk, c) \in d_{\mathrm{A}}(V_A)$ but $(pk, c) \notin D^{T^{(i)}}_{\mathrm{A}}$. By definition of $D^{T^{(i)}}_{\mathrm{A}}$, this probability is at most δ, for each value of $T^{(i)}$. Thus, overall too, the probability of a Lookup failure at any one round is at most δ. By union bound, the probability of any lookup failure occuring is $O(N\delta)$. □

Claim 6. *Suppose Equation 4 holds. In the coupled execution of Π^* and Π, the probability of Extraction failure is $O(N\varepsilon/\delta)$.*

Proof. Extraction failure occurs for Alice if for some $(pk, c) \in D^T_{\mathrm{A}}$, $\mathsf{SimAssist}_B$ fails to find (m, pk, c) such that $\mathsf{Enc}(pk, c) = m$. Note that $\mathsf{SimAssist}_B$ samples a view V'_A conditioned on having a query of the form $[\mathsf{Gen}(sk') = pk]$ in that view. Extraction failure occurs if for this sk', $\mathsf{Dec}(sk', c) = \perp$.

We note that since $(pk, c) \in D_{\mathrm{A}}^{T}$, the test queries $\mathsf{Test}_1(pk)$ and $\mathsf{Test}_2(pk, c)$ are heavy queries (with answer 1) and hence must already be present in the transcript T which includes Eve's messages, and should have the answer 1. Then, we can divide the above failure event into two cases (by considering the view of Bob as well): when Bob's view V_B has a secret key $sk \neq sk'$ and when Bob's view has no secret key for pk. Given the perfect correctness requirement for the oracle, and that $\mathsf{Test}_1(pk) = 1$ and $\mathsf{Test}_2(pk, c) = 1$, the case when Bob has the same secret key $sk = sk'$, but decryption yields $\perp$, does not arise. Note that Bob's view V_B here is with respect to the actual oracle (available to V_A) and not an independently chosen one (as in the case of V'_A).

We analyze the above two cases now.

For the second of the above two cases, note that the event we are considering has it that Alice's view V_A does not contain a Gen oracle call that yields pk (since otherwise the $\mathsf{SimAssist}_B$ would not have reached the third step and aborted) and Bob's view also does not have one, but the test query $\mathsf{Test}_1(pk) = 1$ is present in both their views. This is an extremely unlikely event (see Lemma 3), and its probability can be bounded by a negligible function of κ.

Now we turn to analysing the first case, that is, in some round i Bob's view V_B has a secret key sk different from the secret key sk' present in the sampled view V'_A. This by itself does not give a contradiction (even though Gen is injective), since V'_A was sampled independently from V_B conditioned on T and on having $(pk, c) \in d_{\mathrm{A}}(V'_A)$. Nevertheless we observe the following: given T, if V''_A is picked at random only conditioned on T, then with probability at least δ we have $(pk, c) \in d_{\mathrm{A}}(V''_A)$, and hence the probability of getting $sk' \neq sk$ when sk' is taken from V'_A is at most $1/\delta$ times the probability $sk'' \neq sk$ when sk'' is taken from V''_A. Further, after picking a Bob view V_B, sampling an Alice view V''_A conditioned only on the transcript is identical to drawing $(V''_A, V_B) \leftarrow \mathbf{V}^{(i)}_{A \times B}$. Thus the failure probability using (V''_A, V_B) is at most ε more than the finding two distinct secret keys for pk in $(V_A, V_B) \leftarrow \mathbf{V}^{(i)}_{AB}$. But this last probability is 0. Working backwards, the probability of $sk' \neq sk$ is at most ε/δ.

Thus the probability of extraction failure on receiving a message is $O(\varepsilon/\delta)$ and by union bound, the probability of extraction failure is $O(N\varepsilon/\delta)$. □

To complete our analysis, by the above claims the failure probability in the coupled execution with uniform inputs can be made at most $O(N(\varepsilon/\delta + \delta))$ by using a small enough σ to define Eve. Recall that this experiment picks an input pair uniformly at random. So, for each input pair, the failure probability of the coupled execution is $O(|\mathcal{X}||\mathcal{Y}|N(\varepsilon/\delta + \delta))$, where $\mathcal{X}$ and $\mathcal{Y}$ are the (polynomial-sized) input domains. Then, by Claim 2, the statistical distance between the environment's view in the two executions is $O(|\mathcal{X}||\mathcal{Y}|N(\varepsilon/\delta+\delta))$. Hence by choosing $\varepsilon = \delta^2$ and δ to be $1/(|\mathcal{X}||\mathcal{Y}|N\mathsf{poly})$ for any polynomial poly, we can make the statistical error $O(1/\mathsf{poly})$ (suffering a polynomially related blow up in the communication and oracle-query complexity of the compiled protocol).

7 Common Information Learner for Protocols with Inputs

In this section we shall define the common-information learner for two-party protocols where parties have private inputs. First, we shall describe how the protocols are augmented and, next, we shall prove some properties of the common-information learner (see Lemma 10). This section directly uses the CIL for input-less protocol presented in Lemma 9 to obtain the CIL for protocols with inputs as shown in [MMP12].

7.1 Augmentation of Two-party Protocols

Given a two-party protocol ρ where parties have private inputs, we describe the augmentation algorithm in Figure 9.

1. Given a two-party protocol ρ, define π as the input-less version of ρ, where $x \overset{\$}{\leftarrow} \mathcal{X}$ and $y \overset{\$}{\leftarrow} \mathcal{Y}$.
2. Let π^+ be augmentation of π with Eve_π as defined in Figure 6.
3. The underlying sample space considered will be joint Alice-Bob views as generated by π^+ when the oracle is chosen uniformly at random.

Figure 9: Augmentation algorithm Eve_π for a two-party protocol where parties have private inputs.

7.2 Notation and Definitions

In this section we introduce the following definition:

Definition 6 (Safety). *For Alice view V_A, pair of Bob views (V_B, V_B') and Eve view V_E, we define the following event:* $\mathsf{Safety}(V_A, (V_B, V_B'), V_E) = \mathsf{Nice}(V_A, V_B, V_E) \ \wedge \ \mathsf{Nice}(V_A, V_B', V_E)$.

Analogously, we also define $\mathsf{Safety}((V_A', V_A), V_B, V_E)$.

7.3 Common-Information Learner Properties

Given a 2-party protocol ρ where parties have private inputs, we augment it with Eve_π as described in Figure 9. By suitably choosing the parameter σ for Eve_π, we can ensure the following strong independence properties:

1. Suppose i is an even round in the augmented protocol ρ^+. If $x \in \mathcal{X}$ and $y, y' \in \mathcal{Y}$ are likely inputs at $V_E^{(i)}$ (transcript of the augmented protocol), then the message sent by Alice is nearly independent of Bob's private input being y or y'.
2. Suppose i is an even round in the augmented protocol ρ^+. If $x \in \mathcal{X}$ and $y, y' \in \mathcal{Y}$ are likely inputs at $V_E^{(i)}$, then sample a Alice-Bob joint view $(V_A^{(i+1)}, V_B^{(i)})$ just after Alice has sent the message m_{i+1}. Conditioned on the transcript $V_E^{(i)}$, message sent by Alice m_{i+1} and Bob input being y', sample a new Bob view ${V'}_B^{(i)}$. With high probability: $\mathsf{Safety}(V_A^{(i+1)}, (V_B^{(i)}, {V'}_B^{(i)}), V_E^{(i)})$ holds, i.e. $\mathsf{Nice}(V_A^{(i+1)}, V_B^{(i)}, V_E^{(i)})$ and $\mathsf{Nice}(V_A^{(i+1)}, {V'}_B^{(i)}, V_E^{(i)})$.

3. For an even round i, and likely inputs $x \in \mathcal{X}$ and $y, y' \in \mathcal{Y}$ the distribution of $(V_A^{(i+1)}, V_B^{(i)} {V'}_B^{(i)})$ is close to a product distribution where each component is independently sample conditioned on $(V_E^{(i)}, m_{i+1})$.

Analogous conditions hold when i is odd. The result is formally stated as Lemma 10 and the proof follows from Lemma 9. Note that all these guarantees are only with respect to *likely inputs*. We emphasize that we need to use these properties in our argument in context where the private inputs being considered are all likely; which shall be the case in our setting.

Lemma 10 (Common Information Learner for Likely Inputs)**.** *Function $f : \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{Z}$ is a (symmetric) 2-party deterministic function with polynomial domain size. Let ρ be a normal protocol for secure function evaluation of f relative to an oracle class $\mathbb{O}_\kappa^{(\mathbb{K})}$ (for some key set $\mathbb{K}$) and parties have query complexity at most ψ. Consider Eve_π parametrized by σ and the sample space of Alice-Bob joint views as defined in Figure 9. For every $\varepsilon = 1/\mathsf{poly}(\kappa)$, there exists $\sigma = 1/\mathsf{poly}(\kappa)$ such that: With probability $1-\varepsilon$ over the choice of $V_E \sim \mathbf{V}_E$ (i.e. complete views of Eve_π), the following conditions hold:*

1. *For every even round i and any input $x \in \mathcal{X}$, $y, y' \in \mathcal{Y}$ such that $\mathrm{P}[x, y | V_E^{(i)}] \geq \varepsilon$ and $\mathrm{P}[x, y' | V_E^{(i)}] \geq \varepsilon$, we have:*

$$\mathbf{\Delta}\left((\mathbf{m}_{i+1} | V_E^{(i)}, x, y), (\mathbf{m}_{i+1} | V_E^{(i)}, x, y') \right) \leq \varepsilon.$$

   We remind that $V_E^{(i)}$ is restriction of V_E to first i rounds.

2. *Consider any input $x \in \mathcal{X}$, $y, y' \in \mathcal{Y}$ and even round i. Note that m_{i+1} is already fixed in V_E. Define,*

$$\mathcal{D} \equiv (\mathbf{V}_A^{(i+1)}, \mathbf{V}_B^{(i)} | V_E^{(i)}, m_{i+1}, x, y) \times (\mathbf{V}_B^{(i)} | V_E^{(i)}, V_A^{(i+1)}, m_{i+1}, x, y')$$

   If $\mathrm{P}[x, y | V_E^{(i)}, m_{i+1}] \geq \varepsilon$ and $\mathrm{P}[x, y' | V_E^{(i)}, m_{i+1}] \geq \varepsilon$ then, we have:

$$\mathop{\mathrm{P}}_{(A,B,B') \sim \mathcal{D}} [\mathsf{Safety}(A, (B, B'), V_E^{(i)})] \geq 1 - \varepsilon.$$

3. *Consider any input $x \in \mathcal{X}$, $y, y' \in \mathcal{Y}$ and even round i. Let $\mathcal{D}$ be the distribution as defined above, and $\mathcal{D}'$ be the product distribution:*

$$(\mathbf{V}_A^{(i+1)} | V_E^{(i)}, m_{i+1}, x) \times (\mathbf{V}_B^{(i)} | V_E^{(i)}, m_{i+1}, y) \times (\mathbf{V}_B^{(i)} | V_E^{(i)}, m_{i+1}, y')$$

   If $\mathrm{P}[x, y | V_E^{(i)}, m_{i+1}] \geq \varepsilon$ and $\mathrm{P}[x, y' | V_E^{(i)}, m_{i+1}] \geq \varepsilon$ then, we have:

$$\mathbf{\Delta}\left(\mathcal{D}, \mathcal{D}' \right) \leq \varepsilon$$

Analogous conditions (by swapping Alice and Bob views) also hold for odd i.

Proof. We shall prove this lemma in three parts. We shall show that by choosing σ sufficiently small one can satisfy the conditions 1, 2 and 3 independently. Setting σ to be smaller than all these three respective values of σ we get the above result.

Part 1 and Part 2. These two parts are identical to the theorems proven in Appendix A of [MMP12] using Lemma 9.

Part 3. We will use the following two lemmas.

Lemma 11 (From Section A.2 in [MMP12]). *Let $\mathbf{a}, \mathbf{b}$ be two random variables such that $\boldsymbol{\Delta}(\mathbf{a}, \mathbf{b}) \leq \sigma$. Let E be an event such that $P[a \in E] \geq \delta > 0$ and $P[b \in E] > 0$. Define $\mathbf{a}_E \equiv (\mathbf{a} \mid E)$ and $\mathbf{b}_E \equiv (\mathbf{b} \mid E)$. Then, $\boldsymbol{\Delta}(\mathbf{a}_E, \mathbf{b}_E) \leq \sigma/\delta$.*

Lemma 12. *Let $\mathbf{a}, \mathbf{b}$ be two random variables such that $\boldsymbol{\Delta}((\mathbf{a}, \mathbf{b}), \mathbf{a} \times \mathbf{b}) \leq \alpha$. Suppose $\mathbf{m}$ is another random variable which is a function of $\mathbf{a}$. Then, with probability at least $1 - \sqrt{\alpha}$ over the choice of $m \sim \mathbf{m}$, it holds that the following distributions are all $O(\sqrt{\alpha})$ close:*

$$\mathcal{D}_1 = (\mathbf{a}, \mathbf{b} \mid m), \mathcal{D}_2 = (\mathbf{a} \mid m) \times (\mathbf{b} \mid m), \mathcal{D}_3 = (\mathbf{a} \mid m) \times \mathbf{b}.$$

In particular, for any such choice of $m \sim \mathbf{m}$, it holds that the distribution $\mathbf{b}$ remains $O(\sqrt{\alpha})$ close to $(\mathbf{b} \mid m)$.

Proof. Let $\mathbf{F}$ be a distribution with 3 components defined as follows: We first sample from $(\mathbf{a}, \mathbf{b})$ jointly, and then we sample $\mathbf{m}$ conditioned on $\mathbf{a}$. Let $\mathbf{F}'$ be a similar random variable with the difference that we sample the first two components from $\mathbf{a} \times \mathbf{b}$ independently. First note that the statistical distance cannot increase under the same transformation; and so: $\boldsymbol{\Delta}(\mathbf{F}, \mathbf{F}') \leq \alpha$. Also, since the third component $\mathbf{m}$ is distributed identically in $\mathbf{F}$ and $\mathbf{F}'$, we can expand the statistical-distance expression between $\mathbf{F}, \mathbf{F}'$ over $\mathbf{m}$ as follows.

$$\alpha \geq \boldsymbol{\Delta}\left(\mathbf{F}, \mathbf{F}'\right) = \mathop{\mathbb{E}}_{m \sim \mathbf{m}} \boldsymbol{\Delta}((\mathbf{a}, \mathbf{b} \mid m), (\mathbf{a} \mid m) \times \mathbf{b})$$

By an averaging argument over m we conclude that with probability $1-\sqrt{\alpha}$ over $m \sim \mathbf{m}$, it holds that $\boldsymbol{\Delta}(\mathcal{D}_1, \mathcal{D}_3) \leq \sqrt{\alpha}$. One of the results in Section A.2 in [MMP12] shows that when a distribution (here $\mathcal{D}_1$) is $\delta = \sqrt{\alpha}$-close to a product distribution (here $\mathcal{D}_3$), then it is at most $3\delta = 3\sqrt{\alpha}$ far from the product of its own marginals (here $\mathcal{D}_2$). This finishes the proof the lemma. $\square$

In the following we will first compare the first two components of $\mathcal{D}$ and $\mathcal{D}'$, and then we will compare their last components.

We use Part 3 of Lemma 9 with Eve's parameter equal to σ (which we can choose arbitrarily polynomially smaller than ε). Therefore, in the following we will suppose that *strong* $1-\sigma$ independence already holds over the choice of $V_E^{(i)}$ (which happens with probability $1-\sigma > 1-\varepsilon/10$). Now, using Lemma 12 we conclude that with probability $1-\sqrt{\sigma} > 1-\varepsilon/10$ over the choice of m_{i+1} (which is only a function of $V_A^{(i+1)}$) the joint random variables $(\mathbf{V}_A^{(i+1)}, \mathbf{V}_B^{(i)})$ conditioned on $(V_E^{(i)}, m_{i+1})$ will be $O(\sqrt{\sigma})$-close to the product their marginals. For such choice of $(V_E^{(i)}, m_{i+1})$ we prove the third claim of the lemma by comparing the (first two components of the) distributions $\mathcal{D}$ and $\mathcal{D}'$. Now we use Lemma 11 as follows. Let $\mathbf{a}$ be the joint distribution of of $\mathbf{V}_A^{(i+1)}, \mathbf{V}_B^{(i)}$ conditioned on $V_E^{(i)}, m_{i+1}$ and let $\mathbf{b}$ be the product of the marginal distributions of the two components of $\mathbf{a}$ (i.e. sampling views of Alice and Bob independently). We let E be the event that Alice is using input x and Bob is using y. Since we are assuming the probability of E is at least ε, Lemma 11 shows that (for the fixed pair $V_E^{(i)}, m_{i+1}$) the following two distributions are $O(\sqrt{\sigma}/\varepsilon) < \varepsilon/10$-close:

$$(\mathbf{V}_A^{(i+1)}, \mathbf{V}_B^{(i)} | x, y, V_E^{(i)}, m_{i+1}) \text{ and } (\mathbf{V}_A^{(i+1)} | V_E^{(i)}, m_{i+1}, x, y) \times (\mathbf{V}_B^{(i)} | V_E^{(i)}, m_{i+1}, x, y).$$

But, since the second distribution above is a product distribution the first component does not depend on y and the first component does not depend on x, and therefore it is simply the same as $(\mathbf{V}_A^{(i+1)} | V_E^{(i)}, m_{i+1}, x) \times (\mathbf{V}_B^{(i)} | V_E^{(i)}, m_{i+1}, y)$.

Thus far, we have proved that with probability at least $1 - O(\sqrt{\sigma}) \geq 1 - \varepsilon/10$ over the choice of V_E (which includes $V_E^{(i)}, m_{i+1}$) the first two components of $\mathcal{D}_1$ and $\mathcal{D}_2$ are $O(\sqrt{\sigma})$ close. Now we bound the statistical distance of the third components by showing that with "high" probability over the sampled components V_E (which includes $V_E^{(i)}, m_{i+1}$) and $V_A^{(i+1)}, V_B^{(i)}$, the last components of $\mathcal{D}_1, \mathcal{D}_2$ are also statistically close.

In the following we will be using the same sampled $V_E^{(i)}, m_{i+1}$. We will first show that the following two distributions are close:

$$(\mathbf{V}_A^{(i+1)}, \mathbf{V}_B^{(i)} | x, y', V_E^{(i)}, m_{i+1}) \text{ and } (\mathbf{V}_A^{(i+1)} | V_E^{(i)}, m_{i+1}, x) \times (\mathbf{V}_B^{(i)} | V_E^{(i)}, m_{i+1}, y').$$

Then we will argue, with an averaging argument, that with high probability over the (already) sampled $V_A^{(i+1)}$ the last components of $\mathcal{D}, \mathcal{D}'$ are close. In fact, the very same argument that we used already for inputs (x, y) shows that (for the fixed $V_E^{(i)}, m_{i+1}$) the above two distributions are $O(\sqrt{\sigma}/\varepsilon)$ close. An averaging argument over the sampled $V_A^{(i+1)}$ shows that with probability at least $1 - \sqrt{O(\sqrt{\sigma}/\varepsilon)} > 1 - \varepsilon/10$ over the choice of $V_A^{(i+1)}$ the distribution of $(\mathbf{V}_B^{(i)} | x, y', V_E^{(i)}, m_{i+1})$ (which is the distribution of the last component of $\mathcal{D}$ conditioned on the previously sampled components) will be $\sqrt{O(\sqrt{\sigma}/\varepsilon)} < \varepsilon/10$-close to $(\mathbf{V}_B^{(i)} | V_E^{(i)}, m_{i+1}, y')$ (which is the distribution of the last component of $\mathcal{D}'$ conditioned on the previously sampled components). $\square$

7.4 Extension of [MMP12] to Image-testable Random-oracles

The frontier analysis of [MMP12] is agnostic of the actual oracle class used. It only requires three set of results which are ensured by:

1. Resampling of local views, see Lemma 5,
2. Oblivious Re-randomization, see Lemma 6, and
3. Common-information Learner for protocol with inputs, see Lemma 10.

Thus, extending [MMP12] to the class of (keyed version) of image-testable random oracles.

Lemma 13. *Suppose ρ is a $1 - \lambda(\kappa)$ semi-honest secure protocol (with round complexity N) for 2-party finite semi-honest non-trivial f relative to oracle class $\mathbb{O}_\kappa^{(\mathbb{K})}$, for any key set $\mathbb{K}$. There exists $\Lambda = 1/\mathsf{poly}(\cdot)$ such that, for infinitely many κ, we have $\lambda(\kappa) > \Lambda(N, \kappa)$.*

We emphasize that the insecurity parameter $\Lambda(N, \kappa)$ is *independent* of the query complexity of protocol ρ. The number of additional queries performed by the parties in their curious attacks depends on ψ; but not the insecurity it demonstrates.

8 Putting Things Together

Now we prove the negative result of Theorem 1. Suppose f is a 2-party finite semi-honest non-trivial SFE. Assume that there exists a $1 - \mathsf{negl}(\kappa)$ secure protocol ρ relative to oracle class $\mathbb{PKE}_\kappa$, with round complexity N. By Theorem 6, we construct a new protocol which $1 - \lambda^*(\kappa)$ secure protocol ρ^* relative to oracle class $\mathbb{O}_\kappa^{(\mathbb{K})}$, where λ^* could be arbitrarily small $1/\mathsf{poly}$ and $\mathbb{K} = \{0,1\}^{2\kappa} \cup \{\perp\}$.

By Lemma 13, $\lambda^*(\kappa)$ must be $> \Lambda(N, \kappa)$ for infinitely many κ.

We choose λ^* sufficiently small so that $\lambda^*(\kappa) < \Lambda(N, \kappa)$. This contradicts the above lemma and hence the assumption that ρ is a $(1 - \mathsf{negl}(\kappa))$ secure protocol for semi-honest non-trivial f relative to $\mathbb{PKE}_\kappa$. Note that this result crucially relies on the fact that Theorem 6 preserves round-complexity and the simulation error exhibited in Lemma 13 is function of only round complexity (and independent of the query complexity).

Finally, Corollary 2 restricted to symmetric SFE follows from: a) the result in [MPR09] which shows that the set of symmetric SFE which is standalone or UC trivial in the $\mathcal{F}_{\mathsf{com}}$-hyrbid is identical to the set of semi-honest trivial SFE (which were characterized by [Kus89, Bea89, MPR09]) , and b) the result in [PR08] which shows that standalone or UC-secure protocols for symmetric SFE are also semi-honest secure. Next, extension of this result to general SFE follows from [MPR12].

9 Conclusions and Open Problems

As mentioned in the introduction, our result can be set in the larger context of the "cryptographic complexity" theory of [MPR10]: with every (finite, deterministic) multi-party function f, one can associate a computational intractability assumption that there exists a secure computation protocol for f that is secure against semi-honest corruption. The main result of this work shows that the set of such assumptions associated with 3-party functions is strictly larger than the set associated with 2-party functions. However, *we do not characterize this set either for the 3-party case or for the 2-party case.*

It remains a major open problem in this area to understand what all computational intractability assumptions could be associated with multi-party functions. For the 3-party case, this question is far less understood than that for 2-party functions. Intuitively, there are many more "modes of secrecy" when more than two parties are involved, and these modes will be associated with a finer gradation of intractability assumptions. Our result could be seen as a first step in understanding such a finer gradation. It raises the question whether there are further modes of secrecy for larger number of parties, and if they always lead to "new" complexity assumptions.

Stepping further back, the bigger picture involves randomized and reactive functionalities, various different notions of security, and "hybrid models" (i.e., instead of considering each multi-party function f and a secure protocol for it in plain model, we can consider a pair of functions (f, g) and consider a secure protocol for f given ideal access to g). The cryptographic complexity questions of such functions remain wide open.

References

[Bea89] Donald Beaver. Perfect privacy for two-party protocols. In Joan Feigenbaum and Michael Merritt, editors, *Proceedings of DIMACS Workshop on Distributed Computing and Cryptography*, volume 2, pages 65–77. American Mathematical Society, 1989. 2, 38

[BM07] Boaz Barak and Mohammad Mahmoody. Lower bounds on signatures from symmetric primitives. In *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*, 2007. 2

[BM09] Boaz Barak and Mohammad Mahmoody. Merkle puzzles are optimal - an $O(n^2)$-query attack on any key exchange from a random oracle. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 374–390. Springer, 2009. 1, 2, 5, 10, 43, 44

[BPR$^+$08] Dan Boneh, Periklis A. Papakonstantinou, Charles Rackoff, Yevgeniy Vahlis, and Brent Waters. On the impossibility of basing identity based encryption on trapdoor permutations. In *FOCS*, pages 283–292. IEEE Computer Society, 2008. 2

[DLMM11] Dana Dachman-Soled, Yehuda Lindell, Mohammad Mahmoody, and Tal Malkin. On black-box complexity of optimally-fair coin-tossing. In *Theory of Cryptography Conference - TCC 2011*, 2011. 1

[GGKT05] Rosario Gennaro, Yael Gertner, Jonathan Katz, and Luca Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM J. Comput.*, 35(1):217–246, 2005. 2

[GKM$^+$00] 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. IEEE Computer Society, 2000. 2, 4, 8, 24

[GMR01] Yael Gertner, Tal Malkin, and Omer Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *FOCS*, pages 126–135, 2001. 2

[Hai08] Iftach Haitner. Semi-honest to malicious oblivious transfer - the black-box way. In *TCC*, pages 412–426, 2008. 2

[Hai13] Iftach Haitner. personal communication, Jan. 21, 2013. 5

[HHRS07] Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. Finding collisions in interactive protocols - a tight lower bound on the round complexity of statistically-hiding commitments. In *FOCS*, pages 669–679. IEEE Computer Society, 2007. 2

[HNO$^+$09] Iftach Haitner, Minh-Huyen Nguyen, Shien Jin Ong, Omer Reingold, and Salil P. Vadhan. Statistically hiding commitments and statistical zero-knowledge arguments from any one-way function. *SIAM J. Comput.*, 39(3):1153–1218, 2009. 2

[HOZ12] Iftach Haitner, Eran Omri, and Hila Zarosim. On the power of random oracles. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:129, 2012. 2

[HOZ13] Iftach Haitner, Eran Omri, and Hila Zarosim. Limits on the usefulness of random oracles. *Theory of Cryptography Conference (TCC, to appear)*, 2013. 1, 2, 5, 10

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *FOCS*, pages 230–235. IEEE, 1989. 2

[Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *Structure in Complexity Theory Conference*, pages 134–147, 1995. 2

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In David S. Johnson, editor, *STOC*, pages 44–61. ACM, 1989. 1, 2, 5

[KMR89] Stuart A. Kurtz, Stephen R. Mahaney, and James S. Royer. The isomorphism conjecture fails relative to a random oracle (extended abstract). In David S. Johnson, editor, *STOC*, pages 157–166. ACM, 1989. 13

[KST99] Jeong Han Kim, Daniel R. Simon, and Prasad Tetali. Limits on the efficiency of one-way permutation-based hash functions. In *FOCS*, pages 535–542, 1999. 2

[KSY11] Jonathan Katz, Dominique Schröder, and Arkady Yerukhimovich. Impossibility of blind signatures from one-way permutations. In Yuval Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 615–629. Springer, 2011. 2

[Kus89] Eyal Kushilevitz. Privacy and communication complexity. In *FOCS*, pages 416–421. IEEE, 1989. 2, 38

[LOZ12] Yehuda Lindell, Eran Omri, and Hila Zarosim. Completeness for symmetric two-party functionalities - revisited. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 116–133. Springer, 2012. 4

[LTW05] Henry C. Lin, Luca Trevisan, and Hoeteck Wee. On hardness amplification of one-way functions. In Joe Kilian, editor, *TCC*, volume 3378 of *Lecture Notes in Computer Science*, pages 34–49. Springer, 2005. 2

[MM11] Takahiro Matsuda and Kanta Matsuura. On black-box separations among injective one-way functions. In Yuval Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 597–614. Springer, 2011. 2

[MMP12] Mohammad Mahmoody, Hemanta K. Maji, and Manoj Prabhakaran. Limits of random oracles in secure computation. *CoRR*, abs/1205.3554, 2012. 1, 3, 2, 4, 5, 12, 24, 34, 35, 36, 37

[MPR09] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. Complexity of multi-party computation problems: The case of 2-party symmetric secure function evaluation. In Omer Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 256–273. Springer, 2009. Full version available from IACR Eprint Archive: http://eprint.iacr.org. 2, 38

[MPR10] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. Cryptographic complexity classes and computational intractability assumptions. In Andrew Chi-Chih Yao, editor, *ICS*, pages 266–289. Tsinghua University Press, 2010. 1, 2, 38

[MPR12] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. A unified characterization of completeness and triviality for secure function evaluation. In Steven D. Galbraith and Mridul Nandi, editors, *INDOCRYPT*, volume 7668 of *Lecture Notes in Computer Science*, pages 40–59. Springer, 2012. 2, 38

[Ost91] Rafail Ostrovsky. One-way functions, hard on average problems, and statistical zero-knowledge proofs. In *Structure in Complexity Theory Conference*, pages 133–138, 1991. 2

[OW93] Rafail Ostrovsky and Avi Wigderson. One-way functions are essential for non-trivial zero-knowledge. Technical Report TR-93-073, International Computer Science Institute, Berkeley, CA, November 1993. Preliminary version in Proc. 2nd Israeli Symp. on Theory of Computing and Systems, 1993, pp. 3–17. 2

[PR08] Manoj Prabhakaran and Mike Rosulek. Cryptographic complexity of multi-party computation problems: Classifications and separations. In David Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 262–279. Springer, 2008. 38

[RTV04] Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In Moni Naor, editor, *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2004. 1, 2

[Sim98] Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *EUROCRYPT*, pages 334–345, 1998. 2

A Technical Results

In this section we shall present some technical results needed in the paper.

A.1 Result about Independence

Let G be a bipartite graph with partite sets U and V; and edge set E. Consider a probability distribution $\mathbf{d}$ over E and the probability of an edge (i, j) is represented by $d(i, j)$. Additionally, for $i \in U$, $j \in V$ and $(i, j) \in G$ we have:

$$d(i,j) = c \cdot a(i) \cdot b(j) \cdot (1 + \varepsilon_{i,j})$$

And $|\varepsilon_{i,j}| \leq \varepsilon$. Let $\mathbf{a}$ be probability distributions over U such that the probability of $i \in U$ according to the distribution $\mathbf{a}$ is $c_a \cdot a(i)$, where c_a is a normalization constant. Similarly, define $\mathbf{b}$ the probability distribution over V such that the probability of $j \in V$ is $c_b \cdot b(j)$.

Consider a new distribution $n(i, j)$ over E defined by the following sampling algorithm:

1. Sample $i \sim \mathbf{a}$ and $j \sim \mathbf{b}$.
2. If $(i, j) \notin E$, go to Step 1.
3. Output (i, j).

In this section we shall interpret probability distributions as vectors, where the indices are entries in the sample space. We shall use the fact that:

$$\sum_{(i,j)\in E} ca(i)b(j) \in \left[\frac{1}{1+\varepsilon}, \frac{1}{1-\varepsilon}\right]$$

Otherwise, it cannot be the case that $\sum_{(i,j)\in E} ca(i)b(j)(1 + \varepsilon_{i,j}) = 1$, because $|\varepsilon_{i,j}| \leq \varepsilon$.

Note that $n(i, j) = c' \cdot a(i)b(j)$, where c' is a suitable normalization constant.

$$\begin{aligned} \frac{d(i,j)}{n(i,j)} &= \frac{c(1+\varepsilon_{i,j})}{c'} \\ &= (1 \pm \varepsilon)\frac{c}{c'} = (1 \pm \varepsilon)c\left(\sum_{(i,j)\in E} a(i)b(j)\right) \\ &\in \left[\frac{1-\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\right] \end{aligned} \tag{5}$$

A.2 Reduction in Niceness

In this section we prove the following lemma:

Lemma 14. $b_{i+1} + c_{i+1} \leq c_i + \Theta(\psi^2\sigma) + \mathsf{negl}(\kappa)$, *where* $\sigma = 1/\mathsf{poly}(\kappa)$.

Proof. By definition, for every $V_E^{(i)} \in \mathbb{C}^{(i)}$ we have $\neg\mathsf{Long}(V_E^{(i)})$ and $\mathrm{P}[\mathsf{Nice}(\mathbf{V}_A^{(i)}, \mathbf{V}_B^{(i)}, V_E^{(i)}) | V_E^{(i)}] \geq \frac{1}{2}$.

Since Eve stopped querying, for every every $q \notin \mathcal{Q}(P_E^{(i)})$, we have: $\mathrm{P}[q \in \mathcal{Q}(V_A^{(i)}) \cup \mathcal{Q}(V_B^{(i)})] < \sigma$. Hence, we can claim that: $\mathrm{P}_{(V_A,V_B)\sim\mathsf{Simulate}(V_E^{(i)})}[q \in \mathcal{Q}(V_A) \cup \mathcal{Q}(V_B)] < 2\sigma(1+\mathsf{negl}(\kappa))$. Otherwise, we have the following contradiction:

$$\begin{aligned}
& \underset{(V_A,V_B)\sim\mathsf{Simulate}(V_E^{(i)})}{\mathrm{P}} [q \in \mathcal{Q}(V_A) \cup \mathcal{Q}(V_B)] \geq 2\sigma(1+\mathsf{negl}(\kappa)) \\
\implies \quad & \mathrm{P}[q \in \mathcal{Q}(V_A^{(i)}) \cup \mathcal{Q}(V_B^{(i)}) | \mathsf{Nice}(\mathbf{V}_A^{(i)}, \mathbf{V}_B^{(i)}, V_E^{(i)})] \geq 2\sigma \\
\implies \quad & \mathrm{P}[\mathsf{Nice}^{(i)} | V_E^{(i)}] \times \mathrm{P}[q \in \mathcal{Q}(V_A^{(i)}) \cup \mathcal{Q}(V_B^{(i)}) | \mathsf{Nice}(\mathbf{V}_A^{(i)}, \mathbf{V}_B^{(i)}, V_E^{(i)})] \geq \sigma \\
\implies \quad & \mathrm{P}[q \in \mathcal{Q}(V_A^{(i)}) \cup \mathcal{Q}(V_B^{(i)}) | V_E^{(i)}] \geq \sigma
\end{aligned}$$

Suppose i is even, i.e. Alice is supposed to send the next message in protocol π. Consider the view of Alice *just after* she performs her sequence of queries to the oracle. Let her view be represented by: $V_A^{(i,+)} = (r_A, m^{(i)}, P_A^{(i+1)}, P_E^{(i)})$.

Let μ_{i+1} be the probability that $\mathsf{Nice}(P_A^{(i)} \cup P_B^{(i)} \cup P_E^{(i)})$ but $\neg\mathsf{Typical}(P_A^{(i+1)} \cup P_B^{(i)} \cup P_E^{(i)})$ holds. Intuitively, this accounts for the case that (one of the) the new queries performed by Alice is an atypical query while $P_A^{(i)} \cup P_B^{(i)} \cup P_E^{(i)}$ was typical and good. Note that $\sum_{i\in[n]} \mu_i \leq \mathsf{negl}(\kappa)$, because μ_i is the probability of performing an atypical query when the query complexity of the system is bounded by $2\psi\left(1 + {}^1\!/\!_{\sigma^2}\right)$ (by Lemma 3 and Equation 1).

Let γ_{i+1} be the probability that $\mathsf{Nice}(P_A^{(i)} \cup P_B^{(i)} \cup P_E^{(i)})$ but $\neg\mathsf{Good}(P_A^{(i+1)} \cup P_B^{(i)} \cup P_E^{(i)})$ holds. In other words, we are accounting for the case that the new queries performed by Alice yields an intersection query while $P_A^{(i)} \cup P_B^{(i)} \cup P_E^{(i)}$ was typical and good. This reduces to a claim in [BM09] which shows that:

Claim 7 (Restated from [BM09])**.** $\gamma_{i+1} \leq 2 \times \psi^2 \times 2\sigma(1+\mathsf{negl}(\kappa)) = 4\psi^2\sigma + \mathsf{negl}(\kappa)$.

Next, considering Eve queries after Alice sends a message we have the following cases to consider:

1. If Eve performs too many queries, then the view $V_E^{(i+1)}$ will be accounted in $\mathbb{A}^{(i+1)}$,
2. If $\neg\mathsf{Long}(V_E^{(i+1)})$ then the probability that Eve performs an atypical query can be accounted in ρ_{i+1} and is $\mathsf{negl}(\kappa)$,
3. Otherwise, if $\mathsf{Nice}(P_A^{(i+1)} \cup P_B^{(i)} \cup P_E^{(i)})$ holds then extra Eve queries preserves this property.

Therefore, it follows that $b_{i+1} + c_{i+1} = c_i + \Theta(\psi\sigma(1+\mathsf{negl}(\kappa))) + \mu_{i+1} + \rho_{i+1} \leq c_i + \Theta(\psi\sigma) + \mathsf{negl}(\kappa)$. □

A.3 Product Distribution Basics

Lemma 15. *Relative to oracle class $\mathbb{O}_\kappa^{(\mathbb{K})}$, suppose V_E is an Eve view such that* $\mathrm{P}[\mathsf{Nice}(V_A, V_B, V_E)|V_E] \geq 1-\delta$ *and for any $q \notin \mathcal{Q}(P_E)$ we have* $\mathrm{P}[q \in \mathcal{Q}((P_A \cup P_B) \setminus P_E)] \leq \delta$*. For $\delta = o(1) = 1/\mathsf{poly}(\kappa)$, we have:* $\mathbf{\Delta}\left((V_A, V_B|V_E), (V_A|V_E) \times (V_B|V_E)\right) \leq 3(\psi+1)\delta$.

Proof. Consider the following two distributions:

1. $\mathcal{D}_1$: This is the joint distribution (V_A, V_B) conditioned on the event that V_E occurs. That is, $\mathcal{D}_1$ is the distribution $(V_A, V_B|V_E)$.
2. $\mathcal{D}_2$: This is the joint distribution (V_A, V_B) conditioned on the event that V_E and $\mathsf{Nice}(\mathbf{V}_A, \mathbf{V}_B, V_E)$ occurs.

Note that $\mathbf{\Delta}(\mathcal{D}_1, \mathcal{D}_2) \le \delta$, because $\mathrm{P}_{(V_A,V_B)\sim\mathcal{D}_1}[\mathsf{Nice}(V_A, V_B, V_E)|V_E] \ge 1 - \delta$.

Define distribution $\mathcal{D}_3$ as $\mathsf{Simulate}(V_E)$. We know that $\mathbf{\Delta}(\mathcal{D}_2, \mathcal{D}_3) \le \mathsf{negl}(\kappa)$ (by Lemma 7).

Note that for any $q \notin \mathcal{Q}(P_E)$ we have $\mathrm{P}_{(V_A,V_B)\sim\mathcal{D}_3}[q \in \mathcal{Q}((P_A \cup P_B)\setminus P_E)] \le \delta(1+\mathsf{negl}(\kappa))/(1-\delta)$, because $\mathrm{P}_{(V_A,V_B)\sim\mathcal{D}_1}[\mathsf{Nice}(V_A, V_B, V_E)|V_E] \ge 1 - \delta$, the queries are at most "δ-heavy" in the actual distribution $\mathcal{D}_1$, and $\mathbf{\Delta}(\mathcal{D}_2, \mathcal{D}_3) \le \mathsf{negl}(\kappa)$.

Let $\mathcal{D}_4$ be the distribution over Alice-Bob joint views defined as follows: The probability of Alice-Bob joint view (a, b) has probability $\mathrm{P}_{(\tilde{a},\tilde{b})\sim\mathcal{D}_3}[\tilde{a} = a] \times \mathrm{P}_{(\tilde{a},\tilde{b})\sim\mathcal{D}_3}[\tilde{b} = b]$.

By Barak-Mahmoody-09 [BM09] we know that: $\mathbf{\Delta}(\mathcal{D}_3, \mathcal{D}_4) \le 2\psi \times \delta(1 + \mathsf{negl}(\kappa))/(1 - \delta)$.

Because of $\mathbf{\Delta}(\mathcal{D}_3, \mathcal{D}_1 \equiv (V_A, V_B|V_E)) \le \delta + \mathsf{negl}(\kappa)$, we have: $\mathbf{\Delta}(\mathcal{D}_4, (V_A|V_E) \times (V_B|V_E)) \le 2(\delta + \mathsf{negl}(\kappa))$.

By triangle inequality, we get:

$$\begin{aligned}
\mathbf{\Delta}(\mathcal{D}_1, (V_A|V_E) \times (V_B|V_E)) &\le \mathbf{\Delta}(\mathcal{D}_1, \mathcal{D}_2) + \mathbf{\Delta}(\mathcal{D}_2, \mathcal{D}_3) + \mathbf{\Delta}(\mathcal{D}_3, \mathcal{D}_4) + \mathbf{\Delta}(\mathcal{D}_4, (V_A|V_E) \times (V_B|V_E)) \\
&\le \delta + \mathsf{negl}(\kappa) + 2\psi\delta/(1-\delta) + 2(\delta + \mathsf{negl}(\kappa)) \\
&\le 3(\psi + 1)\delta
\end{aligned}$$

This completes the proof of the lemma. □