ON COMPUTATIONAL INTRACTABILITY ASSUMPTIONS IN CRYPTOGRAPHY

BY

HEMANTA KUMAR MAJI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Doctoral Committee:

Associate Professor Chandra Chekuri, Chair
Associate Professor Manoj Prabhakaran, Director of Research
Assistant Professor Nikita Borisov
Professor P R Kumar
Professor Rafail Ostrovsky, University of California Los Angeles

Abstract

In cryptographic protocols, honest parties would prefer that their security is assured even in presence of adversarial parties who have unbounded computational power. Information theoretic secure realization of cryptographic primitives provides such guarantees; but for most tasks such strong security guarantees cannot be provided for any reasonable notion of security. The standard technique used in cryptography is to assume the existence of some puzzle, whose hard instances are easy to generate but no efficient algorithm can solve them. Such assumptions, which define intractable problems for efficient algorithms, are called *computational intractability assumptions.*

In this work, we motivate a study of computational intractability assumptions which is goal driven and lends support to the fundamental nature of some of the traditional assumptions beyond being historical accidents. Secure multi-party computation deals with the study of constructing secure protocols for general cryptographic tasks which conform to various notions of security. Inspired by complexity theory, we use the notion of *reduction* to further our understanding of computational intractability assumptions. Our framework explores the hardness of natural assumptions of the form: "Task $\mathcal{F}$ can be securely computed given an ideally secure facility for computing $\mathcal{G}$".

Formally, we characterize the *minimal* computational intractability assumption which is sufficient to securely realize a functionality using trusted copies of some other functionality.A functionality $\mathcal{F}$ reduces to $\mathcal{G}$, similar to the complexity-theoretic notion of reduction, if $\mathcal{F}$ can be securely realized when parties can access trusted copies of $\mathcal{G}$. Thus, if $\mathcal{F}$ does not reduce to $\mathcal{G}$ information-theoretically, we are interested in characterizing the minimal assumption, which is sufficient, to securely realize the reduction of $\mathcal{F}$ to $\mathcal{G}$.

In our framework, we explore the relative strengths of these minimal computational assumptions. The fundamental problem, then, is to establish relations, like implication, equivalence and separation, among the assumptions which correspond to various reductions. In this work, we further our understanding of the minimal computational assumptions corresponding to such reductions by showing the following results:

1. We identify a spectrum for the set of assumptions that correspond to reduction between a pair of tasks under different security notions. These reductions are implied by the computational intractability

assumption: “semi-honest secure protocol for oblivious transfer exists” (sh-OT assumption). Additionally, the information theoretically impossible reductions considered in this thesis imply the assumption: “one-way functions exist” (OWF assumption). We conjecture that OWF assumption is necessary for these reductions. In fact, the reductions considered in this thesis are either information theoretically true, false, equivalent to OWF assumption or equivalent to sh-OT assumption [MPR10a].

We expand our study to encompass reductions involving randomized functionalities and study the consequences of providing parties with a trusted source of common unbiased coins. For a strong notion of security they turn out to be useless and unless sh-OT assumption holds these randomness sources are not useful [MOPR11, MP11].

2. We [MPS10] seek to lend additional support to the widely believed premise that “non-trivial cryptography entails the existence of one-way functions”. We show that (constant round) weak coin-tossing protocols, imply OWF assumption. For the general problem, i.e. secure weak coin-tossing protocols with polynomial round complexity, we show a slightly weaker implication $\mathsf{NP} \not\subseteq \mathsf{BPP}$.

3. Finally, we indicate the possibility of a wide range of intractability assumptions in our spectrum, intermediate to OWF assumption and sh-OT assumption. We show that a class of natural assumptions corresponding to reductions that are relativistically separated from OWF assumption. Similar techniques have been used to show their separation from the assumption that “public-key encryption exists” (PKE assumption) [MMP11]; but these assumptions are not known to imply sh-OT assumption. We conjecture that, in fact, sh-OT assumption is relativistically separated from these assumptions.

To my parents.

Acknowledgments

I would like to thank my thesis advisor Manoj Prabhakaran for introducing me to the field of Cryptography and, subsequently, helping me specialize in it. He has kept me busy with a constant supply of intriguing problems and is also a co-author of most of my papers. Working with him has been an extremely enlightening experience and I have learned a lot from the patience he has shown over years of guidance he has provided me.

I have also had the pleasure of collaborating with several prominent researchers in the last few years. I had lots of insightful discussions with Vipul Goyal, Mohammad Mahmoody, Mike Rosulek and Amit Sahai which helped me grow as a researcher. It was an amazing experience working with Vipul during Summer-2010 and collaborating with him over the last year.

The thesis presentation has also benefitted from the valuable inputs of my dissertation committee members Nikita Borisov, Chandra Chekuri, P. R. Kumar and Rafail Ostrovsky. I specially want to thank Chandra for helping me with an initial draft of this thesis and providing suggestions on improving the presentation.

I would also like to thank the faculty and students of the Theory Group of Department of Computer Science, University of Illinois at Urbana-Champaign for a friendly and pleasant stay. Most of the duration of my Ph.D. study was funded by NSF grants CNS 07-47027 and CNS 07-16626 grants.

Finally, I am indebted to my parents for motivating me to pursue my ambitions; and their unwavering and unconditional support over the years. This dissertation is dedicated to them.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Information theoretic cryptography, how-so-ever desirable, is extremely restrictive due to its stringent requirements. In fact, most non-trivial cryptography is information theoretically impossible. For example, even fundamental cryptographic primitives like one-way functions (OWF), public-key encryption (PKE) and key-agreement (KA) over public channels are information theoretically impossible. So, to gain sophistication in cryptographic primitives, we forgo information theoretic security and assume limitations on what can be efficiently computed. If some cryptographic primitive is information theoretically impossible, then we attempt to build secure protocols for it conditioned on some *computational intractability assumption.* In other words, based on the assumption that some computational problem is hard to solve efficiently, a secure protocol realizing the cryptographic primitive is constructed.

This might lure a cryptographer into assuming security of constructions in an ad hoc manner, i.e. one might exhaustively assume that any construction is a secure realization of a particular cryptographic primitive unless proven otherwise. In theoretical cryptography, it is preferred to base the security of protocols on *general* computational intractability assumptions like existence of one-way functions (OWF assumption), existence of public-key encryption (PKE assumption), existence of key-agreement protocols (KA assumption) etc. instead of assuming that some particular protocol is secure. For example, instead of constructing protocols based on the assumption that a particular function, like SHA-1 [Uni95], is one-way, it is preferred to condition the security of the construction on the existence of *any* one-way function[1].

It is possible that some cryptographic primitive can be securely realized conditioned on several computational intractability assumptions. For example, digital signature constructions can be based on the assumption that trapdoor one-way permutations exist [BM88] as well as the existence of one-way functions [NY89, Rom90]. While it is known that existence of trapdoor one-way permutations entails the existence of one-way functions, i.e. if there is a trapdoor one-way permutation then there exists a one-way function [IL89], we do not know whether existence of one-way functions entails the existence of trapdoor one-way permutations; although, there are evidences to the contrary [IR89, GKM$^+$00]. So, it is conceivable that

[1] Levin [Lev85] showed that there is a *universal* one-way function, i.e. if one-way functions exist then he provided construction of a function which is a one-way function.

trapdoor one-way permutations do not exist but one-way functions do. In that case, the latter construction of digital signatures, which is conditioned on OWF assumption continues to be secure while the one based on existence of trapdoor one-way permutations might turn insecure. Thus, given two realizations of a particular cryptographic primitive based on two *different* computational intractability assumptions, we prefer the construction which is based on the *possibly weaker* assumption. Extending this argument, we would prefer the realization based on a *minimal* computational intractability assumption.

This dissertation aims to systematically measure the relative strengths of computational intractability assumptions. Intuitively, we are interested the following general question:

Question 1. *What is the nature of cryptographic complexity necessary and sufficient to implement a cryptographic task.*

Understanding the nature of the computational intractability assumption inherent to a cryptographic task gives us a measure of the complexity of the task. The stronger the computational intractability assumption associated with a task, the more complex the task is. Surprisingly, it is not the case that all different cryptographic tasks have different associated computational intractability assumption. For example, digital signatures and pseudorandom generators, which perform completely different cryptographic tasks, exist if and only if OWF assumption holds [IL89, NY89, Rom90, ILL89, Hås90, HILL99]. In this sense, these computational assumptions are great levelers as they facilitate secure realization of several different kinds of cryptographic tasks. This indicates that these assumption are fundamental to cryptography and by initiating a systematic study of these assumptions, we could explore possible existence of new computational assumptions which are fundamental to cryptography. The following questions informally summarize the guiding problems in this field:

1. Is a computational assumption *possibly weaker* than another computational assumption?

2. Are two computational assumptions *equivalent*?

3. Is one computational assumption *strictly weaker* than another computational assumption? Alternatively, does the former computational assumption *not imply* the latter assumption?

These informal notions mentioned above, respectively, correspond to the complexity theoretic relations: implication, equivalence and separation. Interestingly, computational intractability assumptions do not admit a total order based on their strength, i.e. there are two assumptions such that neither of them implies the other [GKM$^+$00].

In subsequent chapters, we shall introduce a general framework to systematically explore the hardness associated with performing various cryptographic tasks. Abridged versions of the results presented here appear in [MPR10a, MPS10, MOPR11, MMP11, MP11]. Other results relevant to this framework have also appeared in [MPR09, MPR10b] which are explained in greater detail in [Ros09].

1.1 Reduction Model

To motivate the framework used to explore computational intractability assumptions in this dissertation, we use analogies from complexity theory. Reducing arbitrary computational problems to a representative few is a very standard technique in complexity theory. For this purpose, a widely prevalent notion of reduction is *polynomial-time reduction*, because traditionally polynomial-time computations are interpreted as the set of all efficient computations. Let $\textsc{class}(3\text{-}\texttt{SAT})$ be the set of languages which are polynomial time reducible to the problem $3\text{-}\texttt{SAT}$, i.e. given access to an oracle which identifies $3\text{-}\texttt{SAT}$ instances, this set consists of all languages which could be identified by performing polynomial time computation and making polynomially many calls to this oracle. It is known that $3\text{-}\texttt{SAT}$ is NP-complete, and hence $\textsc{class}(3\text{-}\texttt{SAT})$ is identical to the set of all NP problems. Thus, we can conclude that $\textsc{class}(3\text{-}\texttt{SAT})$ contains the sets $\textsc{class}(\texttt{HAM})$ and $\textsc{class}(2\text{-}\texttt{SAT})$. Further, the sets $\textsc{class}(3\text{-}\texttt{SAT})$ and $\textsc{class}(\texttt{HAM})$ are identical; but it is unknown whether $\textsc{class}(3\text{-}\texttt{SAT})$ and $\textsc{class}(2\text{-}\texttt{SAT})$ are identical or not. But there are oracle separation results which show that relative to some oracle, these sets are different [BGS75]; while relative to some other oracle they are identical. Since the exact characterization of what is efficiently computable, i.e. the characterization of the complexity class P or BPP, is unknown, complexity theorists explore the consequences of collapses among such sets of problems. For example, if $\textsc{class}(3\text{-}\texttt{SAT})$ is identical to $\textsc{class}(2\text{-}\texttt{SAT})$ then which other classes would collapse?

In cryptography, instead of problems like $3\text{-}\texttt{SAT}$, $\texttt{HAM}$ etc. the objects of investigation are cryptographic tasks like committing a bit, obliviously transferring a bit, tossing an unbiased coin etc. Similar to the approach in complexity theory, where we are interested in the set $\textsc{class}(L)$ for some language L, the atomic structure of investigation in *cryptographic complexity theory* is the set $\textsc{class}(\mathcal{G})$, where $\mathcal{G}$ is some cryptographic task; though the notion of reduction is slightly modified. The set $\textsc{class}(\mathcal{G})$ consists for all cryptographic tasks which can be efficiently and *securely* performed given access to a trusted $\mathcal{G}$-realizer. Observe that for different notions of security, like semi-honest, standalone and universally composable security, the set $\textsc{class}(\mathcal{G})$ might be different; but fixing a notion of security fully determines the notion of reduction. The collapse of two classes $\textsc{class}(\mathcal{G})$ and $\textsc{class}(\mathcal{G}')$, which are not unconditionally identical, represents an implicit bound on the computational power of the adversaries and, hence, is interpreted as a *computational intractability assumption.* In particular, there can be a protocol for $\mathcal{G}$ reducing to $\mathcal{G}'$ that is secure only if an adversary cannot efficiently perform certain computations. In cryptographic complexity theory, consequences of such collapses are explored. For example, with respect to universally composable security, $\textsc{class}(\mathcal{F}_{\textsc{com}})$ collapses to $\textsc{class}(\mathcal{F}_{\textsc{coin}})$ if and only if $\textsc{class}(\mathcal{F}_{\textsc{ot}})$ collapses to $\textsc{class}(\mathcal{F}_{\textsc{coin}})$, where $\mathcal{F}_{\textsc{com}}$, $\mathcal{F}_{\textsc{coin}}$ and $\mathcal{F}_{\textsc{ot}}$ are, respectively, the commitment, coin tossing and oblivious-transfer functionalities.

Traditionally, cryptographers have been interested in two main questions:

1. For a particular notion of security, which functionalities can be securely realized even against computationally unbounded adversaries? Such functionalities are called *trivial.* Several works have characterized the trivial functionalities in various security models like semi-honest [BGW88, CCD88, RB89, Kus89, Bea89, MPR09, KMR09], standalone [MPR09, KMR09] and universally composable [CKL03, Lin04, PR08] security. Intuitively, trivial functionalities are the easiest class of functionalities with respect to the notion of security being considered.

2. Suppose we assume that some cryptographic intractability assumption is true. Based on this assumption, cryptographers explore the possibility of existence of some additional setup which can help securely compute every functionality. Intuitively, the stronger cryptographic intractability assumption we assume, the larger is the collection of such setups. These attempts are aimed to understand *complete* setups [Yao86, GMW87, Kil88, Kil91, Kil00, CLOS02]. Similar to the analogy mentioned above, these functionalities are hardest to realize with respect to a security notion.

As evident from the discussion, the commonly explored problems lie on the two extremes of a hypothetical spectrum of difficulty of realizing functionalities. It is rare that the set of functionalities, with respect to some notion of security, can be partitioned into trivial and complete functionalities [CK89, BMM99, MPR10b, Kre11]. There are functions with intermediate hardness but the exact measure of their complexity is unknown. A reduction based framework, as introduced by [KMO94] and motivated by analogies from complexity theory, is ideal to explore these *intermediate levels* of hardness.

This dissertation is intended to understand relations like implications, equivalence and separations among various computational intractability assumptions which arise in this reduction based framework. Beyond the elegance of this unifying general framework, the motivation of using a reduction based approach to capture computational intractability assumption also emanates from the fact that these computational assumptions are *natural* to cryptography, i.e. necessary and sufficient to perform certain cryptographic tasks. An aesthetic difference between the reduction framework of complexity theory and the cryptographic complexity theory is that cryptographers believe that these collapses occur, i.e. the associated computational assumption is true, but complexity theorists believe that the reductions among their classes are false. Moreover, for the notions of security studied in this work, there is an upper bound on the strength of computational assumption which are represented by such collapse of classes [MPR10b]. Although this framework is broad, the study we undertake is not exhaustive. There are widely prevalent general computational intractability assumptions in cryptography which are stronger, in a relativistic sense, than the any of the assumptions corresponding to the collapse of various classes in this framework considered in this work. All result considered in this dissertation are restricted to 2-party functionalities, both the functionality that is being realized and the setup functionality, and we shall not be concerned with issue like fairness etc. (unlike [FM00, FGMO05, GIM$^+$10]).

In this section, we shall introduce the simulation based security paradigm and an introduction to the notion of reduction (Section 1.1.1). Finally, we shall conclude with a short discussion on Impagliazzo's worlds [Imp95] and how our work contributes to further the understanding of these worlds (Section 1.1.2).

1.1.1 Ideal-Real Paradigm

In this dissertation we shall use a simulation based definition of security introduced in the seminal work by Goldreich et al. [GMW87]. Intuitively, the security definition can be explained as follows. In the real world, parties have local inputs and communicate to each other via point-to-point private channel trying to securely accomplish some task which is a function of their local inputs. While in the ideal world, parties have access to a *trusted third party* who performs the same task; so the parties could forward their local inputs and receive their respective outputs from the *trusted third party.* The protocol in the real world is secure, if any real world adversarial strategy can be simulated by an ideal world adversarial strategy. The *simulator* in the ideal world mimics the behavior of the adversarial parties such that any environment interacting with the parties, both honest and adversarial, cannot distinguish the real world execution from the ideal world execution.

A protocol is secure if for every adversary in the real world (in which parties execute a protocol), there is an adversary, or *simulator*, in the ideal world that achieves the same effect in every environment. Depending on the nature or adversary/simulator and the environment, we consider three different kinds of security notions.

- A semi-honest (in the real or ideal execution) is one which is not allowed to deviate from the (real or ideal) protocol. Semi-honest security is achieved if for every semi-honest adversary in the real world there is a semi-honest simulator in the ideal world as above.

- A standalone environment is one which does not interact with the adversary during the execution of the protocol. Standalone security is achieved if we restrict the security requirement to standalone environments; in this case the simulator can rewind the adversary without the environment detecting it.

- Universally composable (UC) security [Can01] is achieved when the security requirement is met against all adversaries (possibly active) and all environments (possibly not standalone); the simulator is allowed to be an active adversary. In this case there must exist a straight-line blackbox simulation (i.e., the simulator internally runs the adversary as a blackbox and never rewinds it).

In this work, we exclusively consider *static* adversaries, who do not adaptively corrupt honest parties during the execution of a protocol.

Computationally bounded vs. Computationally unbounded setting. In the computationally bounded setting we restrict all entities of our experiments – the environment, the adversary and simulator – to probabilistic polynomial time computation. In the computationally unbounded setting all these entities can be computationally unbounded. However, for the purpose of the results in this work, one could require the simulator in the computationally unbounded setting to be efficient with blackbox access to the adversary. Then, if a protocol is secure in the computationally unbounded setting, it will be secure in the computationally bounded setting too.

Hybrids. The plain model is a real world in which protocols only have access to a simple communication channel; a hybrid model is a real world in which protocols can additionally use a particular trusted functionality. While hybrid worlds are usually considered only for UC security, we also use the terminology in the setting of standalone security. We note that protocols for *non-reactive* functionalities (i.e., those which receive input from all parties, then give output, and then stop responding) do securely compose even in the standalone security setting.

Reduction. The notion of *reduction* was first introduced by Kushilevitz et al. [KMO94]. We say that a functionality $\mathcal{F}$ *reduces* to a functionality $\mathcal{G}$ if $\mathcal{F}$ can be UC-securely realized in the $\mathcal{G}$-hybrid. In the real world protocol, that parties have access to a trusted implementation of $\mathcal{G}$, in addition to the secure point-to-point communication channel, to securely realize $\mathcal{F}$. Suppose π is a UC-secure protocol for $\mathcal{F}$ in the $\mathcal{G}$-hybrid. Then, parties generate a transcript based on their local views and they can also call the trusted $\mathcal{G}$ implementation. The functionality $\mathcal{G}$ can be any arbitrary functionality, i.e. it need not be a two party function, parties need not play fixed roles while calling $\mathcal{G}$ and, in fact, both parties can provide multiple inputs while performing a call to $\mathcal{G}$.

Security in Hybrid worlds. As mentioned earlier, we shall only consider static corruption of parties, i.e. at the beginning of an execution the adversary announces which party it wants to corrupt and cannot corrupt any further party during the execution of the protocol. To show that a protocol π is a secure realization of $\mathcal{F}$ in the $\mathcal{G}$ hybrid, we need to show that for every adversarial strategy in the $\mathcal{G}$-hybrid there exists a simulator in the ideal world such that any environment is unable to distinguish the real execution from the ideal execution. In this work, we shall restrict ourselves to reductions where both $\mathcal{F}$ and $\mathcal{G}$ are (at most) two party functionalities. Henceforth, we present the security definition restricted to this particular case. Suppose Alice is corrupted by the adversary and Bob is honest. The simulator S_π^A for Alice in the ideal execution, interacts with the adversarial Alice so that no environment can distinguish the real from the ideal execution. The simulator also forwards communication between adversarial Alice and the environment. During this execution, the calls to the $\mathcal{G}$ functionality made by the adversarial Alice is answered by the simulator S_π^A. At some point during the interaction with adversarial Alice, the simulator sends an input x to the ideal functionality $\mathcal{F}$ and receives and answer z. The simulator continues the execution with the adversarial Alice and terminates after generating a complete transcript (we can assume that the adversarial Alice strategy always completes a protocol).

If there exists an efficient S_π^A which can make the ideal execution indistinguishable from the real execution to any environment, then $\mathcal{F}$ is secure in the $\mathcal{G}$-hybrid when Alice is corrupt. Additionally, if there exists an efficient simulator S_π^B which shows that $\mathcal{F}$ is secure in the $\mathcal{G}$-hybrid when Bob is corrupt then π is a secure protocol for $\mathcal{F}$ in the $\mathcal{G}$-hybrid. Intuitively, the existence of a simulator shows that any effect achieved by the adversarial party could be reflected in the ideal world itself. The additional power of the simulator, over what the view of parties in $\mathcal{G}$-hybrid, lies in the fact that it receives the calls to $\mathcal{G}$, i.e. it gets to see the input for each call sent by the adversarial party to $\mathcal{G}$, and it decides the reply to each call. So, for example, when $\mathcal{G}$ is $\mathcal{F}_{\text{COIN}}$, the simulator can determine all the coin outcomes at the beginning of the execution and this could provide additional power to the simulator over the parties in the $\mathcal{G}$-hybrid. Another example is when $\mathcal{G}$ is a function whose input is not a deterministic function of the output received from $\mathcal{G}$. In this case it is not possible to be certain of the input sent to $\mathcal{G}$ just from the output provided by $\mathcal{G}$; on the other hand, the simulator gets the additional information when it sees the query made to $\mathcal{G}$.

1.1.2 Impagliazzo's Worlds

The study of consequence of collapses among complexity classes is a central problem of complexity theory. It is unknown how most complexity classes related to each other and people working in different fields of computer science have different beliefs regarding these relations. For example, people working in combinatorial optimization tend to believe that we live in a world where natural problems are not hard to solve; while cryptographers prefer to believe that we live in a world where there are hard problems which are easy to generate. Our understanding of complexity theory has not been able to resolve what kind of world we reside in. To summarize the current state of the art of our understanding of complexity theory, Impagliazzo introduced the notion of five possible worlds [Imp95]; and there are no known results which refute the possibility of any of these worlds. Each world provides different possible resolutions of several fundamental complexity theoretic problems and there are relativistic realizations of each of these worlds. He exemplified this using the famous "P vs. NP" problem, which is a keystone problem of complexity theory.

In his first world, *Algorithmica*, $\mathsf{P} = \mathsf{NP}$ or something similar in spirit like $\mathsf{NP} \subseteq \mathsf{BPP}$ holds. In this world, any theorem which is polynomial time verifiable can be proven in polynomial time. We know that there exists an oracle relative to which $\mathsf{P} = \mathsf{NP}$, i.e. there is a relativized Algorithmica [BGS75]. In this world, any sophisticated classical cryptography like bit commitment, pseudorandom generators, identification schemes, public key-encryption etc. are impossible but optimization versions of NP problems will have efficient solutions.

In the second world *Heuristica*, NP problems are hard in worst case; although efficiently solvable for problem instances which can themselves be sampled efficiently. There could be instances which are hard to solve but it might not be possible to concentrate non-negligible probability concentration on such instances via an efficient sampling algorithm. Thus, NP problems could turn out be *easy on average*. In particular, the time taken to solve an instance could also depend on the time taken by the sampler which generated the problem instance. If NP turns out to be easy on average, it is unknown whether it will imply that hard problems can be solved efficiently, i.e., say, $\mathsf{NP} \subseteq \mathsf{BPP}$. Recently, Impagliazzo [Imp11] has shown evidence that resolving such a problem might not be easy. He showed that, relative to an oracle, although $\mathsf{distNP} \subseteq \mathsf{avgP}$ [2] but $\mathsf{NP} \not\subseteq \mathsf{BPP}$. But it is known that average case search problems reduce to average case decisional problems [BCGL89]. The first two worlds are favorable to people working in combinatorial optimization, because, in worst case, the hard to solve problem instances are also hard to generate.

Although one-way functions do not exist in the third world *Pessiland*, it is easy to sample hard NP instances. In this world it is hard to generate a pair of hard problem instance and its corresponding witness; but it is easy to generate just the, unsolved version of, hard problem instance. It is still an open problem whether non-trivial cryptography can be performed in Pessiland, because most known non-trivial primitives imply the existence of one-way functions [IL89]. Existence of hard on average problems can be used for generic

[2] Benign algorithms are deterministic machines which refuse to answer on a negligible fraction of inputs; otherwise they correctly identify whether the instance belongs to the language or not. The set of all languages identified by benign algorithms comprises the class avgP [Imp95].

de-randomization [NW88]. They provide a continuum of results relating the hardness of approximating $\mathsf{EXPTIME}$ by a particular circuit class and the implications for generic de-randomization. For example, if $\mathsf{EXPTIME}$ is hard for exponentially large circuits then it implies that $\mathsf{P} = \mathsf{BPP}$; and if $\mathsf{EXPTIME}$ is hard for super-polynomially large circuits then $\mathsf{BPP} \subseteq \mathsf{DTIME}(2^{n^{o(1)}})$. This world is an extremely pessimistic world; where people in both combinatorial optimization and cryptography are unsatisfied; but there are possible consequences for de-randomization of algorithms.

In the remaining two worlds, non-trivial cryptography is possible. The fourth world, *Minicrypt*, does not allow cryptography over public channel, like key-agreement over public channels in presence of an eavesdropping adversary; but one-way functions exist. For concreteness and the purposes of this work, let us assume that public-key encryption, which is equivalent to two-round key-agreement protocol, is not possible in Minicrypt. In this world pseudorandom generators [ILL89, Hås90, HILL99], bit commitment schemes [Nao89], digital signatures [NY89, Rom90, KK05] and zero-knowledge proofs for NP statements[GMW86] exist. Impagliazzo and Rudich [IR89] provided a relativized Minicrypt. There are several relativized separations within Minicrypt. Rudich [Rud91] showed that existence of $(i-1)$-round key-agreement protocols is black-box separated from existence of i-round key-agreement protocol, i.e. there is an relative to which i-round key-agreement protocol exists but no $(i-1)$-round key-agreement protocol exists. Additionally, it is also possible that there are new primitives in Minicrypt, other than the traditional key-agreement primitive, and it might be insufficient to only rely on the existence of one-way functions to securely realize them.

Finally, *Cryptomania* is the most powerful world which permits extremely rich cryptographic primitives. In this world public-key encryption is possible and it is also possible to obliviously transfer a bit from one party to another. Although, there is no explicit bound on the strength of computational intractability assumptions in this world it is possible that there exists an upper bound on the strength of the computational assumption which is sufficient to ensure that all classes in cryptographic complexity theory collapse, for a reasonable notion of reduction.

Cryptographic complexity theory provides a systematic approach to explore intermediate levels of granularity within Minicrypt and Cryptomania by using a sufficiently strong notion of security while considering reductions and analyzing the consequences of the collapse among various classes. This dissertation focusses on the following fundamental aspects of Impagliazzo's Worlds:

1. Can non-trivial cryptography be performed below Minicrypt, i.e. in Pessiland? Alternatively, is existence of one-way functions necessary for secure realization of non-trivial cryptographic tasks?

2. Are there new intermediate worlds within Minicrypt? Are there new worlds which lie strictly within Minicrypt and can their relativistic existence be shown?

3. How high do we need to go in Cryptomania to securely perform any natural task, given some form of setup? For a reasonable notion of security, what is the minimal assumption sufficient to securely perform any task given any setup; unless it is impossible to perform such a task in that particular setup.

1.2 Results

In this section we highlight some of our results which help advance the understanding of cryptographic complexity of performing non-trivial tasks. In Section 1.2.1 we present the intuition of some information theoretic separations shown in [MOPR11]. These information theoretic separations are the first step towards understanding the complexity of reductions, because computational intractability assumptions are of potential use only when a reduction is information theoretically infeasible. In Section 1.2.2 we talk about the implications and equivalences of the reductions mentioned in Section 1.2.1 and several other information theoretic separations presented in [MPR09]. This section contains work which has appeared in [MPR10a, MOPR11, MP11] All these reductions imply existence of one-way functions and, in fact, most of them are equivalent to existence of one-way function or existence of semi-honest secure protocol for oblivious-transfer. Most non-trivial cryptography implies existence of one-way function so it is not surprising that the reductions explored in Section 1.2.2 also imply existence of one-way functions; but the complexity of several weak primitives, like weak coin-tossing etc., is unknown. We present some of our results towards understanding the complexity of weak coin-tossing in Section 1.2.3 [MPS10]. Moreover, in Section 1.2.2 there are several reductions whose complexity seem intermediate to existence of one-way functions and existence of semi-honest secure protocol for oblivious-transfer ($\mathsf{sh\text{-}OT}$). These raise hopes of identifying several new and intermediate complexity assumptions which occur naturally in cryptography. In Section 1.2.4 we present, yet unpublished, evidence that these reductions are black-box separated from several familiar complexity assumptions like one-way functions/permutations, ideal-ciphers, public-key encryption etc. [MMP11].

1.2.1 Information Theoretic Separations

The results covered in this section are based on [MOPR11]. The trusted coin functionality, represented by $\mathcal{F}_{\text{COIN}}$, tosses an unbiased coin and announces the outcome to all the parties. Here we shall consider UC-secure reductions to $\mathcal{F}_{\text{COIN}}$ where adversaries have unbounded computational power and the corresponding notion of reduction is represented by $\sqsubseteq$. Similar separation results have been presented in [MPR09] and detailed presentation of these occur in [Ros09]. In the UC framework extremely trivial functions can be securely realized in the plain model [CF01, CKL03, Lin04, PR08] but any non-trivial setup could be of potential use. Consider any UC non-trivial functionality $\mathcal{G}$, at least $\mathcal{G}$ has a UC secure protocol in the $\mathcal{G}$-hybrid; and potentially more functions could reduce to $\mathcal{G}$. It was unknown how useful or useless a particular setup could be. A source of trusted coins as setup, i.e. $\mathcal{F}_{\text{COIN}}$, is a non-trivial functionality and, thus, could be of potential use to securely realize non-trivial functionalities.

We provide a list of some other frequently occurring functionalities this section:

1. Maximum-evaluation or cut-and-choose function ($\mathcal{F}_{\text{CC}}$): Alice has inputs $\{0, 2\}$ and Bob has inputs $\{1, 3\}$ and $\mathcal{F}_{\text{CC}}$ announces the maximum of the parties' local inputs. A generalization of this functionality is $\mathcal{F}_{\text{CC}}^{i,j}$ where Alice's input space is $\{0, 2, \ldots, 2i - 2\}$ and Bob's input space is $\{1, 3, \ldots, 2j - 1\}$; and the functionality announces the maximum of the two inputs provided by Alice and Bob.

2. XOR-evaluation or simultaneous exchange function ($\mathcal{F}_{\text{EXCH}}$): Alice and Bob have inputs $\{0, 1\}$ and $\mathcal{F}_{\text{EXCH}}$ announces the xor of their respective input bits. A generalization of this functionality is $\mathcal{F}_{\text{EXCH}}^{i,j}$ where Alice and Bob's input spaces are, respectively, $\mathbb{Z}_i$ and $\mathbb{Z}_j$; and the functionality announces the sum of the inputs ($\in \mathbb{Z}_{i+j}$) to both the parties.

Our Contributions. We shall discuss three representative results which exhibit all the major techniques necessary to obtain these impossibility results. It was already known that $\mathcal{F}_{\text{CC}}$ is not UC-securely realizable [CKL03, Lin04, PR08] but it was unknown whether some non-trivial setup like $\mathcal{F}_{\text{COIN}}$ could be useful to securely realize it. If $\mathcal{G}$ is a UC-trivial function, then it is easy to see that $\mathcal{F}_{\text{CC}}$ will not be securely realizable in the $\mathcal{G}$-hybrid; but $\mathcal{F}_{\text{COIN}}$ is not UC-trivial and, hence, $\mathcal{F}_{\text{CC}}$ could possibly have a UC-secure protocol in the $\mathcal{F}_{\text{COIN}}$-hybrid. But, we [MPR09, MOPR11] show that:

Informal Result 1. *There is no UC-secure protocol for $\mathcal{F}_{\text{CC}}$ in the $\mathcal{F}_{\text{COIN}}$-hybrid against adversaries with unbounded computational power.*

Similar to the previous case, $\mathcal{F}_{\text{EXCH}}$ is not UC-trivial but it was unknown whether it could have a UC-secure protocol in the $\mathcal{F}_{\text{COIN}}$ hybrid. The approach required for this varies significantly from the previous result because the XOR function evaluation does not have a unique decomposition [Kus89] and hence the main technical tool used in the previous result does not apply to this setting. In [MOPR11], the following result was shown:

Informal Result 2. *There is no UC-secure protocol for $\mathcal{F}_{\text{EXCH}}$ in the $\mathcal{F}_{\text{COIN}}$-hybrid against adversaries with unbounded computational power. In fact, it does not even have a standalone secure protocol in the $\mathcal{F}_{\text{COIN}}$-hybrid.*

A functionality has bidirectional influence if both Alice and Bob have influence on the outcome, i.e. the functionality is not of the form $\mathcal{F}(x)$ or $\mathcal{F}(y)$. The results mentioned above can be used to show that for any (deterministic) bidirectional $\mathcal{F}$, the reduction $\mathcal{F} \sqsubseteq \mathcal{F}_{\text{COIN}}$ is not true. But if a deterministic $\mathcal{F}$ is not bidirectional, then it already is UC-trivial. Thus, it turns out that $\mathcal{F}_{\text{COIN}}$ is useless for deterministic function evaluation in the UC setting.

But when we extend our study to include randomized functions, there are obviously several UC non-trivial functions which are trivial in the $\mathcal{F}_{\text{COIN}}$-hybrid. Consider the set of functions where one of the parties announces the distribution which will be used to sample the outcome and, subsequently, the parties use the $\mathcal{F}_{\text{COIN}}$ functionality to generate public coins which shall be used to sample the outcome from the distribution locally. There functions are essentially of the form: $(x, D_x(r))$, where $D_x(r)$ is the distribution corresponding to Alice input x and then using the random coins r to sample from it. We shall call these functions *publicly-selectable source*, i.e. the distribution from which the output is sampled is publicly known. Most functions which are publicly-selectable source are UC non-trivial and any function which is publicly-selectable source is trivial in the $\mathcal{F}_{\text{COIN}}$-hybrid.

First, unsurprisingly, it can be shown that if a randomized $\mathcal{F}$ has bidirectional influence then $\mathcal{F} \sqsubseteq \mathcal{F}_{\text{COIN}}$ is not true [MOPR11] and the details of this result will be covered in Chapter 4. Functions with no bidirectional influence are called *selectable source*, because one of the parties privately samples the outcome based on her input and announces the outcome. Observe that for any publicly-selectable source function, the output of the function determines the input used to sample the distribution. So, any function which is selectable source but not publicly-selectable source satisfies the following property. There exists two inputs x and x' such that the support of the distributions $f(x)$ and $f(x')$ intersect but the two distributions $f(x)$ and $f(x')$ are not identical. We call such functions *oblivious sampling*, because there exists an output for which one of the parties is not certain which input was used by the other party to sample it. Surprisingly, oblivious sampling functions do not have UC secure protocols in the $\mathcal{F}_{\text{COIN}}$-hybrid and hence:

Informal Result 3. *Unless $\mathcal{F}$ is a publicly-selectable source, $\mathcal{F}$ has no UC-secure protocol in the $\mathcal{F}_{\text{COIN}}$-hybrid against adversaries with unbounded computational power.*

1.2.2 Implications and Equivalences

In this section, we shall explore the complexity of reductions which could possible be true when we consider adversaries with bounded computational power. There reductions are referred to as *conditionally true* reductions, while reductions which hold even in presence of adversaries with unbounded computational power are referred to as *true* reductions. Consider the task of securely realizing a functionality $\mathcal{F}$ in the

$\mathcal{G}$-hybrid with respect to a particular security notion. If the reduction is conditionally true, then the strength of the reduction refers to the computational intractability assumption which is necessary and sufficient to realize the reduction, i.e. securely realize $\mathcal{F}$ in the $\mathcal{G}$-hybrid with respect to the particular security notion using minimal computational intractability assumption. The strength of the reduction could increase if the associated security requirements are strengthened. Strengthening the security notion, thus, leads to the possibility that reductions, which were equivalent with respect to a weaker notion of reduction, become separated when the notion of reduction is strengthened. On the other hand, if we strengthen the notion of reduction by a significant amount, we might cause the reduction to become false as shown in [MPR10b]. So, we need to strike a balance between what we consider an acceptably strong notion of reduction which can exhibit sufficient diversity in computational intractability assumptions associated with reductions as well as most reductions are either true or conditionally true. In this section, we shall consider universally composable security (under static corruption) [Can01] which satisfies both the requirements mentioned above; and we will restrict our scope of study to deterministic functions.

We first mention a result which upper bounds the computational intractability assumption associated with any conditionally true reduction:

Lemma 1 (Maximal Assumption[MPR10b, Ros11]). *When $\mathcal{F}$ and $\mathcal{G}$ are deterministic functionalities, all reductions of the form $\mathcal{F} \sqsubseteq \mathcal{G}$ with respect to universally composable security is: a) true, b) implied by existence of semi-honest secure protocol for oblivious transfer, or c) false. And the only reductions which are false are where $\mathcal{F}$ is UC non-trivial and $\mathcal{G}$ is UC trivial.*

Recall, that UC trivial deterministic functions are extremely simple functions where the output of the functionality is a function of only one party's input. Now, we need to explore the exact characterization of the computational complexity associated with these reductions when $\mathcal{G}$ is non-trivial.

Most of these reductions are not true, i.e. against adversaries with unbounded computational power they do not hold. But, for our results where we show that these reduction imply OWF assumption, we need to simulate these attack if OWF assumption is false. It has been shown that one-way functions and distributionally one-way functions are equivalent [ILL89], hence if OWF assumption is false then distributionally one-way functions also do not exist. If distributionally one-way functions do not exist then the universal generation problem for NP statements [JVV86, BGP00] can be solved efficiently [Ost91, OW93]. And efficient solution of this problem is sufficient to simulate our attacks with a small non-negligible error which can be driven down to arbitrarily low precession.

Our Contribution. It was shown in [MPR09] that when we consider UC-reduction with static corruption, against adversaries with unbounded computational power, there is intricate structure in the landscape of cryptographic complexity. This notion of reduction will be represented as $\sqsubseteq$. These arguments involved exhibiting that any protocol securely realizing a particular function needs to have the following two properties: a) Based on the information leaked, we can identify frontiers in the protocol tree and b) These frontiers much be encountered in some particular order on any root to leaf path in the protocol tree.

Kushilevitz and Beaver [Kus89, Bea89], independently, characterized the class of all symmetric two party deterministic functions which can be semi-honest securely realized. There functions, called decomposable functions, have optimal perfectly secure protocols and the number of rounds of these canonical protocols is called the depth of decomposition tree of the corresponding function. Functions with unique canonical semi-honest protocol are called uniquely decomposable functions. It was shown in [MPR09], that if adversaries have unbounded computational power and both $\mathcal{F}$ and $\mathcal{G}$ are uniquely decomposable, then $\mathcal{F} \not\sqsubseteq^{\text{STAT}} \mathcal{G}$ if $\mathcal{F}$ has greater depth of decomposition tree than $\mathcal{G}$'s depth of decomposition tree. For example $\mathcal{F}_{\text{CC}}^{i,i} \sqsubseteq^{\text{STAT}} \mathcal{F}_{\text{CC}}^{j,j}$ if and only if $j \geq i$. We show that, instead of unbounded computational power, only the assumption that OWF assumption is false suffices. A complementary result from [MPR10b] shows that $\mathcal{F}_{\text{CC}} \sqsubseteq^{\text{STAT}} \mathcal{G}$, if $\mathcal{G}$ has unique decomposition and if OWF assumption is true then $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{F}_{\text{COM}}$, for any $\mathcal{F}$. Thus, we show the following result:

Informal Result 4. *Let $\mathcal{F}$ and $\mathcal{G}$ be uniquely decomposable two-party symmetric deterministic function evaluation such that the decomposition depth of $\mathcal{F}$ is more than the decomposition depth of $\mathcal{G}$. Then $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ is equivalent to* OWF *assumption.*

Further, we also leverage the fact that OWF assumption is false to show the following result:

Informal Result 5. *Let $\mathcal{F}$ be passive trivial but not standalone trivial and $\mathcal{G}$ be standalone trivial but not UC trivial. Then the reduction $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ is equivalent to* OWF *assumption.*

Using the result of [MPR10b], we can show that OWF assumption implies that $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$. For the other direction, suppose π is protocol of $\mathcal{F}$ in the $\mathcal{G}$-hybrid and, since $\mathcal{G}$ is standalone-trivial, $\mathcal{G}$ has a standalone secure protocol ρ. By a result of [PR08] we can show that π with every instance of $\mathcal{G}$ replaced by the ρ protocol gives a protocol π' which is also standalone secure. Now, we need to launch an attack on this protocol which violates its standalone security. There are two cases to consider. If $\mathcal{F}$ is not uniquely decomposable then $\mathcal{F}_{\text{EXCH}}$ can be standalone securely realized in the $\mathcal{F}$-hybrid. We provide an algorithm which correlates the output obtained by the two parties if parties have unbounded computational power. Next, we show that this attack can also be simulated if OWF assumption is false. On the other hand, if $\mathcal{F}$ is uniquely decomposable then the attack of [MPR09] can also be simulated on $\mathcal{F}$, assuming OWF assumption is false.

For our next set of results, we need to introduce a class of function evaluations called exchange-like functionalities. These functionalities allow parties to simultaneously exchange their inputs, like $\mathcal{F}^{i,j}_{\text{EXCH}}$. We show that it is extremely hard to realized any $\mathcal{F}$ in $\mathcal{G}$-hybrid, where $\mathcal{G}$ is exchange-like, unless $\mathcal{F} \sqsubseteq^{\text{STAT}} \mathcal{G}$. Due to the result by [MPR10b], we already know that any exchange-like functionality is complete if sh-OT assumption is true. So, as a consequence, any $\mathcal{F}$ has secure protocols in $\mathcal{G}$-hybrid if sh-OT assumption is true when $\mathcal{G}$ is UC non-trivial. For the other direction, we show that if $\mathcal{F}$ is not exchange-like then either $\mathcal{F}_{\text{OT}}$ or $\mathcal{F}_{\text{CC}}$ reduces to $\mathcal{F}$ statistically. This would imply that $\mathcal{F}_{\text{OT}} \sqsubseteq^{\text{PPT}} \mathcal{G}$ or $\mathcal{F}_{\text{CC}} \sqsubseteq^{\text{PPT}} \mathcal{G}$ and both of them imply that sh-OT assumption is true. But, if $\mathcal{F}$ is itself exchange-like but $\mathcal{F}$ does not statistically reduce to $\mathcal{G}$, then we show that sh-OT assumption must hold. Consequently, we show the following result:

Informal Result 6. *If $\mathcal{F} \sqsubseteq \mathcal{G}$, where $\mathcal{G}$ is exchange-like, then either $\mathcal{F} \sqsubseteq^{\text{STAT}} \mathcal{G}$ or* sh-OT *assumption must hold.*

The same result is also true when $\mathcal{F}$ is a two-party randomized function evaluation and $\mathcal{G}$ is $\mathcal{F}_{\text{COIN}}$.

1.2.3 Weak Coin Tossing

Although, any non-trivial cryptographic task is believed to imply existence of one-way functions, for some weak primitives it is unclear whether one-way functions are necessary for their secure realization. One such weak primitive is *weak coin-tossing* or coin tossing with preferences. This primitive was first proposed by Blum [Blu82] who motivated it using the following real-world problem:

> Alice and Bob are getting divorced and they wish to figure out who gets to keep their favorite car. Both of them have moved on with their lives. Alice, now, has moved to the East coast and Bob has moved to the West coast; and they are refusing to be in the same room at the same time. Thus, they decide to toss a coin by running a protocol over telephone to toss a coin such that the probability of the outcome being Heads or Tails is exactly $1/2$. At the end of the protocols, if the outcome is Heads, then Alice keeps the car; other Bob keeps the car.

Figure 1.1: Weak coin-tossing: Definition.

We emphasize that Alice and Bob both *want* the car, i.e. Alice, if she decides to be malicious, will *only* try to bias the outcome towards Heads. A good protocol for weak coin-tossing will ensure that none of the parties can force its preferred outcome with probability more than $1/2$. If a two party protocol π, between participants party 1 and party 2, satisfies either one of the following two conditions then it can be used for weak coin-tossing:

1. Party 1 cannot bias the outcome towards Heads and party 2 cannot bias the outcome towards Tails, or

2. Party 1 cannot bias the outcome towards Tails and party 2 cannot bias the outcome towards Heads.

If one-way functions exist, then the following 3-round protocol is a good weak coin-tossing protocol:

1. Alice tosses a perfectly random bit $x \leftarrow \{0, 1\}$ and sends a commitment [Hås90, HILL99, Nao89, GL89] of x to Bob.
2. Bob tries to guess x and sends his guess y to Alice.
3. Alice opens the commitment to x.

Bob wins the game if he correctly guesses x; otherwise Alice wins. If any party aborts without completing the protocol then the other party wins.

Figure 1.2: Weak coin-tossing: Blum's Protocol [Blu82].

It is easy to see that if one-way functions exist then this is a good weak coin-tossing protocol where any party can gain, at most, negligible advantage in biasing the outcome towards its preferred outcome. But, are one-way functions necessary for the existence of good weak coin-tossing protocols? To analyze this question, let us state what exactly qualifies as "powerful adversarial attacks" against purported weak coin-tossing protocols. For a protocol π, if we are able to show existence of four adversarial strategies $\mathcal{A}_{1,H}$, $\mathcal{A}_{1,T}$, $\mathcal{A}_{2,T}$ and $\mathcal{A}_{2,H}$ such that:

1. Party 1 using strategy $\mathcal{A}_{1,H}$, while interacting with honest party 2, is able to bias the outcome towards Heads by an additional significant amount; or party 2 using strategy $\mathcal{A}_{2,T}$, while interacting with honest party 1, is able to bias the outcome towards Tails by an additional significant amount, and

2. Party 1 using strategy $\mathcal{A}_{2,H}$, while interacting with honest party 2, is able to bias the outcome towards Tails by an additional significant amount; or party 2 using strategy $\mathcal{A}_{2,T}$, while interacting with honest party 1, is able to bias the outcome towards Heads by an additional significant amount.

If such attacks exist, then it is easy to see that Alice or Bob will be able to force its preferred outcome with probability significantly more than $1/2$, thus the protocol is not a secure weak coin-tossing protocol. We will specify what "significant advantage" means in the following paragraph.

Prior State-of-the-art. Suppose Alice and Bob do not mind if the probability of their preferred outcome remains at least $1/2 - 1/1000$, i.e. any party gets the car with probability at least a small constant below $1/2$.

Thus, "significant advantage" corresponds to achieving more than a small constant advantage, say $1/1000$. Surprisingly, as recently mentioned in [Imp09], nothing significant was known about the complexity of the assumption: "There exists good weak coin-tossing protocols" for such a definition of "significant advantage". To understand the exact complexity of weak coin-tossing, we need to show existence of powerful adversarial strategies conditioned on complexity theoretic assumptions. And we will prefer to use as weak an assumption as possible to launch powerful attacks against any protocol, thus exactly characterizing the complexity of good weak coin-tossing protocols. For concreteness, consider the following set of results which were known prior to our work:

1. If $\mathsf{PSPACE} \subseteq \mathsf{BPP}$, then one of the parties can force its preferred outcome with certainty: Suppose party 1 is trying to bias the outcome towards Heads and party 2 is trying to bias the outcome towards Tails. Since the parties can efficiently solve the complete game tree, because $\mathsf{PSPACE} \subseteq \mathsf{BPP}$, party 1 can force the outcome Heads or party 2 can force the outcome Tails. Thus, we can define strategies $\mathcal{A}_{1,H}$ and $\mathcal{A}_{1,T}$. Similarly, we can also define the strategies $\mathcal{A}_{2,H}$ and $\mathcal{A}_{2,T}$. Thus, Alice or Bob will be able to force its preferred outcome with certainty.

2. If one-way functions do not exist, then one of the parties can force its preferred outcome with non-negligible advantage: If one-way functions do not exist, then a modification of the argument in [CI93] suffices to show that Alice or Bob will be able to force its preferred outcome with probability $1/2 + \Theta(1/\sqrt{r})$, where the protocol has r-rounds. When r is a constant, this implies that one of the parties can obtain a constant advantage. Our results will significantly improve the result for constant r.

3. If $\mathsf{NP} \subseteq \mathsf{BPP}$ and the protocol has constant number of rounds, then one of the parties can force its preferred outcome with certainty: If $\mathsf{NP} \subseteq \mathsf{BPP}$ then $\mathsf{PH} \subseteq \mathsf{BPP}$ [Zac86]; and the complete game tree can be solved efficiently because the game tree has constant depth. Thus, similar to the first result mentioned above, we can claim that one of the parties will be able to force its preferred outcome with certainty. Our results will weaken the complexity assumption and still ensure similar guarantees for the adversarial strategies.

Our Contribution. The above mentioned adversarial strategy due to [CI93] deviates from the honest behavior only once. In a private communication [Imp10], Impagliazzo mentioned that such attack strategies are bound to fail due to tight $1/\sqrt{r}$ bounds for fail-stop adversaries implied by [MNS09]. Thus, any hopes of achieving constant advantage when $r = \omega(1)$ will rely on adversarial strategies which attack, i.e. deviate from the honest behavior, at least $\omega(1)$ times. In a recent work [MPS10], we provide adversarial strategies which attack at every round when they are supposed to send the next message in the protocol and show the following two results:

Informal Result 7. *If $\mathsf{NP} \subseteq \mathsf{BPP}$, then one of the parties can force its preferred outcome with probability at least $3/4$.*

Informal Result 8. *If one-way functions do not exist and the protocol has constant number of rounds, then one of the parties can force its preferred outcome with certainty.*

Informal Result 8 improves the adversarial strategy by [CI93] and the brute-force search strategy when $\mathsf{NP} \subseteq \mathsf{BPP}$. Informal Result 7 is incomparable to the result by [CI93] but improves the brute-force search strategy implied by $\mathsf{PSPACE} \subseteq \mathsf{BPP}$.

1.2.4 Intermediate Assumptions

In this section, we restrict ourselves to two-party symmetric deterministic functions with constant output alphabet size and, as an introductory presentation, we shall limit ourselves to passive corruption. Two-party symmetric deterministic functions can be represented by a matrix whose (i, j)-th entry represents the output of the function when the first party has input i and the second party has input j. Kilian's characterization [Kil91] states that if a function $\mathcal{F}$ has an embedded OR in its function matrix then it is complete, i.e. all other functions information-theoretically semi-honest securely reduce to $\mathcal{F}$. Kushilevitz [Kus89] and Beaver [Bea89] characterized the class of two party functions which have perfectly semi-honest secure protocols. They called such functions decomposable and recently [MPR09, KMR09] showed that even when statistical security is considered, the characterization of triviality remains identical. Surprisingly, for binary output alphabet triviality and completeness are complementary notions [KMO94] and, using Kreitz's [Kre11] characterization of ternary output functions, triviality and completeness are also complementary for ternary output alphabet. But if the output alphabet size is at least 4, then there are functions which are neither trivial nor complete, i.e. there are functions of intermediate complexity. The function tables for such functions with output alphabet size 4 and 5 are provided below:

$$\begin{pmatrix} 1 & 1 & 4 & 3 \\ 4 & 2 & 2 & 3 \\ 4 & 3 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 2 \\ 4 & 5 & 2 \\ 4 & 3 & 3 \end{pmatrix}$$

[Kus89, Bea89, KMO94]

Figure 1.3: Intermediate Functions: Output alphabet size 4 and 5.

Due to a result by [MPR09] it is known that the computational intractability assumption $\mathcal{C}_{\mathcal{F}}$ = "There exists a semi-honest secure protocol for $\mathcal{F}$" implies existence of one-way functions, for any semi-honest non-trivial function $\mathcal{F}$. Moreover, it is trivial to see that existence of a semi-honest oblivious-transfer protocol implies $\mathcal{C}_{\mathcal{F}}$ for any $\mathcal{F}$. But, for an intermediate function $\mathcal{F}$, what is the relation of the assumption $\mathcal{C}_{\mathcal{F}}$ with respect to other familiar computational intractability assumptions like: existence of one-way functions/permutations, public-key encryption and semi-honest oblivious transfer protocols? We wish to highlight that we assume that the input domains of both parties for the function $\mathcal{F}$ grows at most as a polynomial in the security parameter. If super-polynomial inputs domains are permitted then the characterization of two-party symmetric deterministic semi-honest trivial functions by [Kus89, Bea89, MPR09, KMR09] breaks down.

Our Contribution. We merge and generalize the techniques introduced in [MPR09] and [BM09] to show the following result:

Informal Result 9. *For functions with intermediate complexity, the assumption $\mathcal{C}_{\mathcal{F}}$ is black-box separated from* OWF *assumption. It has been recently shown [MMP11] that this assumption is also separated from* PKE *assumption.*

It is still an open problem to show whether semi-honest oblivious-transfer is black-box separated from $\mathcal{C}_{\mathcal{F}}$ or not. We conjecture that this separation holds. The result presented above is a restriction of the actual result proven in [MMP11]. In fact we show that random oracle is as useful as a commitment functionality when semi-honest, standalone and universally composable security are consider. The details of the construction will be provided later [MMP11], in the following paragraphs we highlight some interesting facets of our most basic construction.

The basic result in [MMP11] shows that a random oracle is not useful for semi-honest secure computations of two-party functions. The idea is to show that if there exists a semi-honest secure protocol using random oracle, then we can construct an alternate protocol in the plain model, i.e. parties do not have any access to a random oracle. Simulating access to a random oracle in the plain model is a non-trivial task. Results by [IR89, BM09, DLMM11] can be interpreted as construction of an *independence learner* which removes dependencies between Alice and Bob views. The main hurdle in simulating a random oracle is that we need to answer queries consistently; otherwise Alice and Bob could simulate the part of the random oracle in their local view honestly given a PSPACE Oracle. Another alternative is that only one party simulates the random oracle and the other party asks the first party to answer the oracle queries on her behalf. But the first party learns all the queries that the second party makes and, thus, might make the original protocol insecure.

An independence learner [BM09, DLMM11] solves this problem by capturing all intersection queries, i.e. queries which are common to Alice and Bob's local views, with high probability. The independence learner is a deterministic algorithm and depends only on the public view, i.e. a subset of both Alice and Bob's local views. So, it can be simulated by either Alice or Bob and could be published in the transcript because it reveals information which either party could have generated, thus revealing it will not harm the security of the protocol. Now, any query not yet answered in the public list of query-answers produced by the independence learner are locally answered by the parties themselves. The event that we have a query inconsistently answered by the parties is extremely low because intersection queries are covered with high probability. The construction of such a deterministic and efficient independence learner is provided in [BM09, DLMM11].

Next, we look at some issues which are specific to our problem statement and were absent from the problem considered by [BM09, DLMM11]. Let us highlight a feature of protocols in the plain model: When Alice is supposed to send the next bit in the protocol, she sends it as a deterministic function of her view, which is comprised of her local input x, her local random tape r_A and the partial transcript τ generated so far. In particular, the probability of her next bit being 0 is a function of x and the transcript τ only. We shall call this property "Markov-chain property". To convert a protocol in the random oracle model into a protocol in the plain model, we will need to ensure that the Markov-chain property holds. Let us be slightly more formal about the construction. Suppose Alice and Bob have generated the partial transcript τ and our eavesdropper makes additional queries to the random oracle and the (public) set of query-answer pairs be represented by I. Now, given τ, I and x, if we are able to predict the probability of next message being 0 then we shall be done. The first hurdle here is the fact that *after* the set I is generated by the eavesdropper, Alice queries the random oracle as instructed by the protocol; and *then* deterministically generates the next message as a function of her local view. Querying the random oracle after the augmented transcript τ and I is fixed creates the risk of generating additional correlations between Alice and Bob views and the probability of the next message being 0 might not be a function of x, τ and I only. We should clarify that the eavesdropper is never provided the local inputs of Alice and Bob. All we need to show is that the probability of the next message is a function of local input x and the augmented transcript τ and I, i.e. Alice querying the random oracle after the augmented transcript is fixed has negligible effect. Now, the final hurdle is that the eavesdropper does not know the exact inputs of Alice and Bob[3], i.e. the algorithm to kill dependence between Alice and Bob views should be oblivious of x and y and still remain efficient.

So, we need to construct an efficient eavesdropper strategy which simultaneously resolves the following

[3] The eavesdropper might learn some information about x and y from the transcript itself; but beyond that it is not explicitly provided the values of x or y.

three issues: a) Kill dependence between Alice and Bob views so that the random oracle can be efficiently simulated, b) Ensure the Markov-chain property so that the next message functions can be appropriately defined, and c) Efficiently perform both these task when the local inputs of Alice and Bob, x and y, are not provided explicitly to the independence learner algorithm. The unified solution to all these problems is to run several eavesdropper algorithms. For every input pair $(x, y) \in X \times Y$, define $\pi(x, y)$ as the protocol where Alice and Bob inputs are fixed to x and y respectively. We spawn one eavesdropper algorithms for every $\pi(x, y)$ and it queries all highly likely queries assuming Alice and Bob inputs are x and y respectively. It is not evident whether such an eavesdropper is efficient; but we modify [BM09] to show the efficiency of our eavesdropper algorithm. After all these three properties are satisfied, we can convert this protocol into a secure protocol for $\mathcal{F}$ in the plain model. For this transformation, we shall use the fact that Alice and Bob local views are independent given the augmented transcript. Parties can simulate a random oracle using the following strategy: a) All eavesdropper query-answers are public knowledge and could be generated by a particular party, say Alice, and b) Answer to any query which is outside the public query list generated by the eavesdropper can be locally sampled by the party. The second step might create conflicts, i.e. Alice and Bob might sample answers to the same query locally which are inconsistent, but the probability that such an intersection query is not already covered in the public query-answer list generated by the eavesdropper is extremely small [BM09, DLMM11].

1.3 Organization

In Chapter 2 we summarize prior results relevant to our study (Section 2.1) and introduce definitions and notations (Section 2.2) useful in the subsequent sections. We present our information theoretic separation results in Chapter 3. We shall characterize reductions which are false against adversaries with unbounded computational power when parties can access a secure implementation of unbiased public coins. In Chapter 4, we shall show that several reductions are equivalent to either OWF assumption or $\mathsf{sh\text{-}OT}$ assumption. Chapter 5 presents the consequences of secure weak coin tossing protocols. We show that constant round weak coin tossing protocols imply OWF assumption and if the protocols have polynomial round complexity then they imply $\mathsf{NP} \not\subseteq \mathsf{BPP}$. We discuss evidences supporting the possibility of several intermediate complexity assumptions intermediate to OWF assumption and $\mathsf{sh\text{-}OT}$ assumption in Chapter 6. Finally, we conclude by highlighting some important open problems and conjectures in Chapter 7.

Chapter 2

Prior Work and Preliminaries

In this chapter we will summarize prior results relevant to our study and introduce some useful notations and definition for subsequent chapters. In Section 2.1 we will summarize prior results in multi-party computation and relating strengths of computational intractability assumptions. Finally, in Section 2.2 we will introduce some terminology and definitions to make our subsequent chapters describing our results more accessible.

2.1 Prior Work

In this section we shall present some known results in multi-party computation and establishing relations among various computational assumptions. These fields are extremely vast and it is impossible to cover every significant result relevant to our study. We present an overview of some of the representative results of multi-party computation in Section 2.1.1 focussing on information theoretic reductions, completeness and some results in the computational setting. Most of these results may not be the best possible or most efficient reductions with respect to round complexity, communication complexity etc. Covering such aspects is beyond the scope of this work. In Section 2.1.2 we survey some results which show the relation among various computational assumptions. The section covers prior results which show implications and equivalences of these assumpitons and, finally, separations among them.

2.1.1 Quick Summary of Multi-party Computation Results

Yao posed the following problem, henceforth referred to as the Millionaires' problem [Yao82a]: Two millionaires wish to find out which one of them is richer without divulging any additional information. Formally, suppose their respective assets are worth x and y; and we are interested in computing the function $f(x, y) = 1$ if and only if $(x \leq y)$. Surprisingly, this deceptively simple problem cannot even be resolved when $x, y \in \{0, 1\}$ for any reasonable notion of security. But, if it is guaranteed that $x \neq y$, i.e. say, x is always even and y is always odd, and we are willing to disclose the assets of the richer millionaire then this function can be semi-honest securely computed [Kus89, Bea89, MPR09, KMR09]. On the other hand, if the millionaires reside in a highly concurrent environment, there are stronger security models which rule out secure protocols for this function [CKL03, Lin04]. The study of secure realization of various tasks, possibly interactive ones, is the central focus of *multi-party computation*. Some major representative results from the field of multi-party computation are highlighted in this section and we shall, simultaneously, interpret these results in our reduction based framework.

Most of these results consider a special type of functionalities called *function evaluation*. These extremely fundamental functionalities are non-interactive tasks where parties provide inputs and receive their respective outputs from the trusted third party. Moreover, the functions considered here are restricted to *symmetric functions*, i.e. all parties receive the same output. The function outcomes can be either deterministic or randomized. For deterministic functions, $f(x_1, x_2, \cdots)$ represents the evaluation of the function f where the i-th party has input x_i. When considering randomized functions, $f(x_1, x_2, \cdots)$ represents a distribution over the output alphabet set and the outcome is drawn according to this distribution. The study of symmetric function evaluation is pivotal to better understand any arbitrary function evaluation [KM11, MOPR11].

Information Theoretic Triviality. The main question here is to characterize functions which can be computed in information theoretic secure manner for different adversarial strategies. These results can be interpreted as characterization of functionalities which reduce to the secure-channel functionality, i.e. point-to-point or broadcast channel. Recall that we always assume existence of such channels in our framework and, thus, we represent it using the symbol $\emptyset$. So, we characterize functions which belong in $\textsc{class}(\emptyset)$ for various notions of reduction.

The most basic corruption model is semi-honest corruption, i.e. adversaries follow the protocol honestly but, after the completion of the protocol, all corrupt parties "gossip" and, depending on their own private inputs and outputs received, try to figure out additional information about the honest parties' inputs. A function is t-private if there exists a semi-honest secure protocol where the adversary can corrupt any (up to) t-parties but it cannot violate the security guarantees of the honest parties. Traditionally, symmetric functions, i.e. the functions where all parties receive the same output, are considered and, henceforth, all results in this section, unless otherwise specified, are for symmetric functions. The general question in this line of research can be summarized by the following:

Question 2 (IT-triviality)**.** *Given a n-party function evaluation, determine whether it can be information theoretically t-privately computed in various corruption models, like semi-honest, active or Byzantine, composable etc.; alternatively, solve the membership problem for* $\textsc{class}(\emptyset)$*.*

It was shown by [BGW88, CCD88] that if honest parties are in a majority then any function, which has finite output alphabet, can be semi-honest securely computed even against adversaries with unbounded computational power. This result is tight because it can easily be seen that unless honest parties are in a majority, there are no information theoretically secure protocols for some functions. They showed that every n party function with finite output alphabet set is $\lfloor {}^{n-1}/_2 \rfloor$-private. The assumption that the output set is finite is necessary because [CGK90, CGK95] showed that, even when the output alphabet set is countable but infinite, there are functions which are not even 1-private[1]. Note that for the special case of $n = 2$, the results by [BGW88, CCD88] do not imply anything non-trivial. Kushilevitz [Kus89] and, independently, Beaver [Bea89] characterized 2-party functions which can be passive securely computed with perfect security. They showed that only *decomposable* functions, where each party alternately rule out some of their inputs, are the only 2-party symmetric functions which have perfect semi-honest secure protocols. Recently [MPR09, KMR09] showed that this characterization also extends to statistical semi-honest security.

When the output alphabet size is 2, [CK89] showed that any n-party function is either $\lfloor {}^{n-1}/_2 \rfloor$-private or n-private. They show that any function which is $\lceil {}^{n}/_2 \rceil$-private has a simple n-private semi-honest secure protocol: Every party announces a bit which is dependent on its local input and the output of the function is (mod two) addition of every party's bit. This result can be interpreted as a "zero-one" law, i.e. every function can be characterized into two class. Only recently, [Kre11] extended this result to functions with ternary output alphabets. The n-private protocol for this case is not as trivial as the binary output case, and interested readers can refer to [Kre11] for the details. Any hope that this simple "zero-one" law extended to arbitrary, but constant, output alphabet size was crushed by [CGK94]. They showed that for every n and $t \geq \lfloor {}^{n-1}/_2 \rfloor$, there exists a n-party function with output alphabet size c_n, a constant depending on n, which is t-private but not $(t+1)$-private[2]. In particular, it implies that for $n = 4$, there is a function with output alphabet size 8 which is 2-private but not 3-private (recall that [BGW88, CCD88] guarantees that every function is 1-private for $n = 4$). This shows that for $n = 4$, the "zero-one" law does not hold when the output alphabet set size is 14; and the minimum output alphabet size for which this "zero-one" law fails is unknown.

A stronger model of corruption considers Byzantine adversaries or active adversaries, i.e. adversaries may behave arbitrarily. Similar to the case of semi-honest security, we can consider security against a

[1] There is an interesting, and perhaps unexpected, relation between boolean output functions and circuit lower bounds. The class of functions with linear circuit size is exactly identical to the class of functions which have 1-private protocols and use a constant number of random bits [KOR96]. A generalization of this result [CKOR97] relates the circuit size of a function and the amount of randomness needed to compute it t-privately, for any $t < {}^{n}/_2$.

[2] A common technique used in all the negative results of [CK89, CGK94, CS95] is *partition argument*, i.e. partition the set of parties into two disjoint sets and obtain a secure protocol for a 2-party function from the original protocol for the function. It was unknown whether all such negative results can be obtained by suitably applying the partition argument. Finally, [CI01] showed that partition arguments are not powerful enough for such negative results. They show that as the number of partitions k is increased, the power of the partition argument increases. In particular, they show that there are functions which has a k partition such that the resulting function is fully private; while for all $k' < k$ partitions the resulting functions are not fully private.

Byzantine adversary which gets to control t out of n parties and tries to gain additional information about the private inputs of the honest parties which would otherwise be inaccessible to them. Protocols which are secure even when the adversary corrupts up to t parties are called standalone t-private protocols. For active security, [BGW88, CCD88] show that if $t < n/3$ then every function can be t-privately computed. These results, like their corresponding results for passive corruption, are also tight because even secure channel with guaranteed message delivery cannot be implemented unless $t < n/3$ [LSP82]. But if we assume that an additional broadcast channel is provided to the parties then every function has a secure protocol if $t < n/2$ [RB89][3]. Due to a lower bound by [Dol82], we know that the result of [RB89] is also tight. Both these results were generalized by [HM97] who show that the results by [BGW88, CCD88, RB89] extend to arbitrary sets of adversarial players. They show that if there are no two sets of adversarial players whose union is the set of all parties then there exists a semi-honest secure protocol for any function for that collection of adversarial sets. Similarly, if there are no three sets of adversarial players whose union is the set of all parties then there exists a standalone secure protocol for any function for that collection of adversarial sets. They also provide a result analogous to [RB89] when a broadcast channel is also provided to the parties, though the protocol is not guaranteed to be polynomial time. Instead of general broadcast channels, it has been shown that broadcast between three parties is sufficient to securely compute any functions if $t < n/2$ parties are corrupt [FM00]. Similar completeness results where parties can use a functionality which involves $k < n$ parties is explored in [FGMO05].

Canetti proposed a significantly stronger notion of security: Universally Composable security [Can01], which ensures security of protocols in highly concurrent environments and in presence of arbitrary computations. Although this model of security is extremely demanding and only functions where one of the parties forwards its input to the other have secure protocols [CF01, CKL03, Lin04, PR08]. They, additionally, show that if $\mathcal{F} \notin \textsc{class}(\emptyset)$ then the reduction "$\mathcal{F}$ reduces to $\emptyset$" is false. Despite this strong negative result, non-trivial hybrids seem to possess significantly higher power, for example any non-trivial functionality is trivial in its own hybrid. More discussion on this topic will be included in the following paragraphs.

Completeness. It is useless to provided parties access to a functionality which is information theoretically trivial, because the parties can compute the function on their own from scratch. On the other extreme there are functionalities whose access is sufficient to securely realize any other functionality. These functionalities, following the nomenclature in complexity theory, are called *complete* functionalities. In our framework, $\mathcal{G}$ is a complete functionality if every functionality can be securely realized in the $\mathcal{G}$-hybrid, i.e. all functionalities

[3] Interestingly, the protocols by [BGW88, CCD88] are perfectly secure protocols, whereas the protocols by [RB89] incur a negligible error. It can be shown that, it is impossible to obtain a result similar to [RB89] with perfect security even when parties have access to broadcast channels [BGW88, CCD88, RB89].

lie in $\textsc{class}(\mathcal{G})$. Oblivious transfer is an extremely useful primitive to help realize other functionalities. There are several equivalent variants of oblivious transfer. In the first version, as introduced by [Rab81], the sender has an input bit b and it may or may not be delivered to the receiver with certain probability; but the sender does not know whether the bit was delivered or not. Rabin OT is equivalent to an erasure channel and [Wie83] showed how it could be implemented quantum mechanically. In the second version introduced by [EGL85], called (1-out-of-2) OT, the sender has two bits x_0 and x_1 and the receiver has a choice bit b. The receiver obtains x_b as her output, while the sender gets no output. Observe that it is trivial to convert this functionality into a symmetric functionality by allowing the receiver to send an additional masking bit z and the output $x_b \oplus z$ is announced to both the parties. There are two important generalizations of (1-out-of-2) OT: a) Firstly, the sender has inputs $x_0, x_1, \ldots, x_n$ and the receiver has an input $i \in [n]$, and b) Secondly, the inputs of the sender could be strings instead of bits. All these variants of oblivious transfer are, in fact, equivalent [BCR86, Cré87].

It is easy to see that (1-out-of-n) OT is semi-honest complete for functions with small input domains for both parties. Suppose a party has input x, then it inputs $f(x,0), f(x,1), \ldots, f(x,n)$ to (1-out-of-n) OT and the other party uses $y \in [n]$ as his inputs to obtain $f(x,y)$. Similarly, they can reverse their roles and let the first party learn the outcome of the function. For arbitrary computations, [GMW87, GV87] show that oblivious transfer is semi-honest complete. When the alphabet size is binary or ternary, any n-party function is either trivial, i.e. $\lfloor {}^{n-1}/_{2} \rfloor$-private, or complete [CK89, Kre11]. Such dichotomy exhibited by functions is frequently referred to as "zero-one" law.

Kilian [Kil88] shows that for two party case, Rabin OT suffices to perform oblivious circuit evaluation even against Byzantine adversaries. For two-party symmetric deterministic function evaluation, Kilian [Kil91] shows that if there exists an embedded OR minor[4] then $\mathcal{F}$ is complete. Kushilevitz [Kus89], and independently Beaver [Bea89], characterized two-party symmetric deterministic functions which have perfectly secure protocols. This characterization, called *decomposability*, also extends to the statistical security case [MPR09, KMR09]. Observe that the "zero-one" characterization of [CK89, Kre11] does not imply anything non-trivial for the two party case. Kilian et al. [KMO94, KKMO00] show a similar result for the two party case. They introduce the term reduction to capture secure realization of a function $\mathcal{F}$ in a $\mathcal{G}$-hybrid and show that, for binary output, two-party symmetric deterministic functions are either trivial or complete. Recently,

[4] A functionality $\mathcal{F}$ implementing a function evaluation f has an OR minor if there exists two Alice inputs i_0, i_1 and two Bob inputs j_0, j_1 and two output symbols k_0, k_1 such that $f(i_a, j_b) = k_{a \vee b}$. Alternatively, one can interpret the function f as a two-dimensional matrix where the (i,j)-th entry corresponds to the outputs of the function when Alice has input i and Bob has input j. If there are two columns and two rows, such that the restriction of the matrix to them has three identical output symbols then f has an embedded OR minor.

Kreitz's [Kre11] result implies that this characterization also extends to ternary output alphabets[5]. There are examples of function with output alphabet size 4 with are neither complete nor trivial, thus the "zero-one" law breaks down. The "zero-one" characterization of functions being either trivial or complete also extends to two-party functions where only one of the parties receives the output [BMM99]. Kilian [Kil00], subsequently, provided completeness characterization for asymmetric and randomized function. Surprising, there is a correspondence between arbitrary two-party functions and two-party symmetric functions. Any incomplete deterministic function, it may be non-trivial, is equivalent[6] to a two-party symmetric function [KM11].

Computational Model. For the Millionaires' problem, it can be easily shown that in the information-theoretic model that it is impossible to have a protocol which securely computes this function even when we restrict ourselves to semi-honest corruption.If we are willing to make computational intractability assumptions, then existence of semi-honest secure oblivious-transfer protocol could be used to evaluate the circuit of any function 1-privately via *garbled circuits* [Yao86]. A detailed proof of correctness for the construction is provided in [Gol04, Rog91, LP09]. For the multi-party case, in a seminal work, Goldreich, Micali and Wigderson [GMW87] show that any function can be computed n-privately, even against Byzantine adversaries, if there exists a semi-honest secure protocol for oblivious transfer. The main idea is to semi-honest securely perform an evaluation-under-the-wraps similar to the garbled circuit construction of [Yao86]. They use the Barrington's model [Bar86] to transform any arbitrary computation as a sequential composition of a permutations. To consider Byzantine adversaries, two additional techniques are used. Firstly, a coin-tossing-in-the-well technique is used to generate a honest local random tape for every party if at least one the parties is honest. Next, at every step of the execution, parties provide a zero-knowledge proof [GMR85, GMW86] that they have followed the semi-honest strategy honestly. For more stringent security requirement, like Universally Composable security [Can01], Canetti et al. [CLOS02] showed that under suitable computational assumption the coin-tossing functionality is complete. Recently [MPR10b], it was shown that every Universally-Composable non-trivial function is complete if there exists a semi-honest protocol for oblivious-transfer. Although, in this case only static corruption is considered.

[5] Note that Kreitz's result is only meaningful for $n > 2$. But the main combinatorial characterization shown by him implies this result for the special case of $n = 2$.

[6] A functions f is *extremely trivial* in the g-hybrid if f can be securely computed by: Making a call to the g-oracle followed by some local computation by both parties. Two functions f and g are equivalent if f is extremely-trivial in the g-hybrid and vice-versa.

2.1.2 Computational Intractability Assumptions

In this section we mention some relations among computational intractability assumptions. The results covered in this section show implications, equivalences and separations among various computational assumptions like existence of one-way functions/permutations, key-agreement protocols, public-key encryption etc.

Implications and Equivalences. Primitives like secret-sharing [Sha79, Bla79], verifiable secret-sharing [CGMA85, BGW88, CCD88] and one-time pads are information theoretically possible; but, as mentioned earlier, most non-trivial cryptography is information theoretically impossible. In a seminal work, Impagliazzo and Luby [IL89] showed that many non-trivial cryptographic primitives like private-key (symmetric-key) encryption, identification/authentication, bit-commitment and coin-tossing over the telephone imply existence of one-way functions. They show that any secure protocol for these non-trivial Cryptographic tasks implies the existence of an identification protocol; which, in turn, implies the existence of *distributionally one-way functions*. Surprisingly the notion of distributionally one-way functions is equivalent to the notion of standard notion of one-way functions [ILL89]. Existence of non-trivial zero-knowledge proofs [GMR85], which are extremely useful in cryptography, also imply existence of one-way functions [Ost91, OW93]. Thus, existence of one-way function is possibly weaker than the assumption that cryptographic primitives like private-key encryption, identification schemes, bit-commitment, non-trivial zero-knowledge etc. exist. In other words, existence of these non-trivial cryptographic primitives imply the existence of one-way functions.

Subsequently, Håstad, Impagliazzo, Levin and Luby [ILL89, Hås90, HILL99] showed that pseudorandom generators can be constructed based only on the assumption that one-way functions exist[7]. Naor and Yung [NY89] showed how digital signatures can be constructed from universal one-way hash functions and presented a construction of universal one-way hash functions from one-way permutations. Finally, Rompel [Rom90] constructed universal one-way hash functions based solely on existence of one-way functions (a partial result in this direction appears in [SY90]); thus, providing a construction of digital signatures based on the assumption that one-way functions exist. Interested readers can refer to [KK05] for an alternate proof of the result by [Rom90] which fills several gaps in the original proof. It is also known that one-way functions are sufficient for bit-commitment [Nao89] and zero-knowledge proofs for languages in NP [GMR85, GMW86, IY87, ILL89, Nao89, Hås90]. These results [ILL89, IL89, Nao89, NY89, Hås90, Rom90, Ost91, OW93] show that existence of diverse cryptographic primitives like private-key encryption, pseudorandom generators, bit commitment, digital signatures, non-trivial zero-knowledge are all equivalent to the assumption that one-way

[7] Although the construction of pseudorandom generators based on one-way Permutations was known earlier [Yao82b], the problem of only using one-way functions was open for a long time.

functions exist.

Recall that [CGK94, CI01] showed separations based on the partition arguments. It is noteworthy that the assumption that any of their functions has a semi-honest secure protocol is equivalent to the complete assumption that semi-honest secure protocol for oblivious transfer exists. There are several cryptographic primitives, like public-key encryption, key-agreement protocols over public channels secure against eavesdroppers, semi-honest secure oblivious transfer protocols etc., whose existence entails the existence of one-way functions [IL89]. It is interesting to note that existence of public-key encryption is equivalent to the existence of 2-round key-agreement protocols; and k-round semi-honest secure oblivious transfer protocols entails the existence of k-round key-agreement protocols [GKM^{+}00]. As evident from the discussion above, it appears to be the case that one-way functions are necessary for non-trivial cryptography. In general, the following question is of fundamental interest to cryptographers:

Question 3 (Necessity of one-way functions)**.** *Does the existence of any non-trivial cryptographic construct, which is information theoretically impossible, entail the existence of one-way functions?*

Separations. Unlike implications, it is extremely difficult to define when an assumption is strictly stronger than another. To this end, in a seminal paper, Impagliazzo and Rudich [IR89] introduced the notion of black-box separations. They show that the existence of key-agreement protocols is black-box separated from existence of one-way permutations, i.e. there exists an oracle relative to which one-way permutations exist but key-agreement protocols do not exist. This technique rules out any relativizing construction, in particular black-box constructions, of key-agreement protocols from one-way permutations. The construction in [IR89] uses a *random oracle*[8] and a PSPACE oracle. Although non black-box techniques in complexity theory [Coo71] and cryptography [Yao86, GMW87, Bar01] are highly infrequent, this technique does not rule out the existence of such constructions. Thus, we define an assumption to be strictly stronger than another assumption if there exists a oracle relative to which the latter assumption holds but not the implication. There are several possible variations of defining what qualifies as a black-box construction [RTV04]. The technique used by [IR89] rules out *fully black-box* reductions according to the nomenclature used in [RTV04], i.e. not only the weaker primitive is used as an oracle, even the proof of correctness uses the adversary against the stronger primitive in a black-box manner. Until recently [Bar01] the code of the adversary was not used in any construction.

Subsequent to the work by Impagliazzo and Rudich [IR89], there has been several results showing separation of computational intractability assumptions using similar methodologies. Recently, [BM09] have

[8] Constructions in random oracle world are closely related to constructions in the Ideal-cipher model [CPS08, HKT11]. More details on this aspect will be discussed later.

provided an alternate and tighter proof for the result in [IR89]. Rudich [Rud91] extended this approach to show that existence of $(k-1)$-round key-agreement is black-box separated from the existence of k-round key-agreement, i.e. it is not possible to construct a $(k-1)$-round key-agreement protocol by using a k-round key-agreement protocol in a black-box manner. Simon [Sim98] showed that existence of collision-intractable hash functions [Dam87] are black-box separated from existence of one-way functions. Even one-way permutations are black-box separated from one-way functions [Rud88, KSS00, KSS11] and recently [MM11] show that they cannot be constructed from injective length-increasing one-way function in a black-box manner even if the increase in length is only one bit. Gertner et al. [GKM$^+$00] showed that existence of key-agreement and oblivious-transfer protocols are in some sense incomparable; but existence of oblivious-transfer protocols is black-box separated from existence of public-key encryption protocols. They show that k-round oblivious transfer implies the existence of k-round key-agreement protocols; but even 2-round key-agreement, i.e. public-key encryption, cannot be used in black-box manner to construct any k-round oblivious-transfer protocols. On the other hand, they also show that $(k-1)$-round key-agreement protocols are black-box separated from k-round oblivious-transfer protocols. Gertner et al. [GMR01] show that trapdoor-functions are black-box separated from the existence of trapdoor-predicates like public-key encryption using a weaker notion of separation. In an extension of the Impagliazzo-Rudich [IR89] methodology, Gennaro et al. [GT00, GGK03, GGKT05] prove tight lower bounds for construction of pseudorandom-generators [BM84, GL89], universal one-way hash functions [NY89], encryption and (semantically-secure) signature schemes which match with the best known constructions; otherwise $\mathsf{P} \neq \mathsf{NP}$.

Recently, there have been additional separation results. Gertner et al. [GMM07] show that chosen-ciphertext secure public-key encryption is black-box separated from semantically-secure public-key encryption, if the chosen-ciphertext construction's decryption algorithm does not query the semantically-secure public-key encryption's encryption algorithm. Boneh et al. [BPR$^+$08] show that identity-based encryption [Sha84] is black-box separated from trapdoor-permutations and even chosen-ciphertext secure public-key encryption. Vahlis [Vah10] shows that existence of trapdoor-functions under correlated inputs, introduced in [RS09], is black-box separated from trapdoor-functions. Blind-signatures, introduced by [Cha82], cannot be constructed from one-way permutations in a black-box manner [KSY11].

In general, one of the most fundamental problems in this field of research can be formulated as follows:

Question 4 (IT-Irreducibility vs. Separation)**.** *For any $\mathcal{F}$ which does not information-theoretically (say, semi-honest) reduce to a, possibly non-trivial, functionality $\mathcal{G}$, can the assumption "there exists an information-theoretic (semi-honest) secure protocol for $\mathcal{F}$" be black-box separated from the assumption "there exists an information-theoretic (semi-honest) secure protocol for G"?*

Note that the non-triviality lies in the fact that $\mathcal{G}$ itself is non-trivial, so constructing an oracle which allows only secure computation of $\mathcal{G}$ and not $\mathcal{F}$ is difficult. If an extremely powerful oracle is chosen then it can also help securely realize $\mathcal{F}$ as well. Thus, it is imperative to provide the oracle with intermediate power so that only $\mathcal{G}$ can be securely realized and not $\mathcal{F}$. This implies that we need to have an extremely fine-grained understanding of the complexity of securely realizing functionalities.

2.2 Preliminaries and Definitions

We say that a function $\nu : \mathbb{N} \to \mathbb{R}$ is *negligible* if for every polynomial p, $\nu(k) < 1/p(k)$ for sufficiently large k. If $\mathcal{D}, \mathcal{D}'$ are discrete probability distributions with support S, we write $\mathrm{SD}(\mathcal{D}, \mathcal{D}')$ to denote the statistical distance of the distributions, defined as $\mathrm{SD}(\mathcal{D}, \mathcal{D}') = \frac{1}{2} \sum_{s \in S} |\mathcal{D}(s) - \mathcal{D}'(s)|$.

Security. We use standard conventions and terminology for the security of protocols for multi-party computation tasks. A protocol is secure if for every adversary in the real world (in which parties execute a protocol), there is an adversary, or *simulator*, in the ideal world (in which the task is carried out on behalf of the parties by a trusted third party called a *functionality*) that achieves the same effect. A *semi-honest* or *passive* adversary is one which is not allowed to deviate from the protocol. *Standalone* security is achieved if the simulator is allowed to rewind the adversary; *Universally composable (UC)* security [Can01] is achieved if the simulation is straight-line (i.e., never rewinds the adversary). In this work, we exclusively consider *static* adversaries, who do not adaptively corrupt honest parties during the execution of a protocol.

The *plain model* is a real world in which protocols only have access to a simple communication channel; a *hybrid model* is a real world in which protocols can additionally use a particular trusted functionality. While hybrid worlds are usually considered only for UC security, we also use the terminology in the setting of standalone security. We note that protocols for *non-reactive* functionalities (i.e., those which receive input from all parties, then give output, and then stop responding) do securely compose even in the standalone security setting.

2.2.1 Functionalities

We focus on classifying several important subclasses of functionalities.

Secure function evaluation (SFE). A 2-party secure function evaluation (SFE) functionality is specified by two functions $f_1 : X \times Y \to Z$ and $f_2 : X \times Y \to Z$, where X and Y are finite sets. The functionality waits for input $x \in X$ from Alice and $y \in Y$ from Bob, then delivers $f_1(x, y)$ and $f_2(x, y)$ to them, respectively.

There is no fairness guarantee: if a party is corrupt, it can obtain its own output first and decide whether the output should be delivered to the other party.

If $f_1 = f_2$ are identical we call it a *symmetric* SFE (or SSFE) functionality. SSFE functionalities are the most fundamental, and have been studied since Yao first introduced the concept of multi-party computation [Yao82a]. We can specify an SSFE function by simply giving its function table, where the rows correspond to an input of Alice, and columns correspond to an input of Bob. For instance, the XOR functionality has function table $\boxed{\begin{smallmatrix}0&1\\1&0\end{smallmatrix}}$.

Isomorphism. $\mathcal{F}$ and $\mathcal{G}$ are isomorphic[9] if either functionality can be UC-securely realized using the other functionality by a protocol that is "local" in the following sense: to realize $\mathcal{F}$ given $\mathcal{G}$ (say), each party maps its input (possibly probabilistically) to inputs for the functionality $\mathcal{G}$, calls $\mathcal{G}$ once with that input and, based on their private input, the output obtained from $\mathcal{G}$, and possibly private random coins, locally computes the final output, without any other communication. It is easy to see that isomorphism is an equivalence relation.

Output renaming. Suppose a two-party deterministic SSFE $\mathcal{F}$ implements a function f. We will rename outputs of f which are not identical. For example, consider the function $f : \mathbb{Z}_2 \times \mathbb{Z}_2 \to \mathbb{Z}_2$ where $f(x, y) = x \oplus y$. the output 1 received when the input pair is $(1, 0)$ is different from the output 1 received when the input pair is $(0, 1)$ because both parties can distinguish these two outputs based on their respective local views. More formally, for distinct $x, x' \in X$ and $y, y' \in Y$, if $f(x, y) = z$ and $f(x', y') = z$ but $f(x', y) \neq z$ and $f(x, y') \neq z$ then these two instances of output z can be renamed to separate outputs. For every possible function, we rename the outputs such that there no such renaming is possible. For example, the function $\mathcal{F}_{\text{EXCH}}$ represented by $\boxed{\begin{smallmatrix}0&1\\0&1\end{smallmatrix}}$ can be renamed to $\boxed{\begin{smallmatrix}1&2\\3&4\end{smallmatrix}}$. Note that output renaming keeps the new function isomorphic to the original function.

Usefulness of a source. We say that a source of common randomness $\mathcal{G}$ is **useless** in realizing a 2-party functionality $\mathcal{F}$ if either $\mathcal{F}$ could be securely realized in the plain model (i.e., without using $\mathcal{G}$) or $\mathcal{F}$ cannot be securely realized even in the $\mathcal{G}$-hybrid model. Note that we consider only the feasibility question and not any efficiency issues.

[9] The definition given here is a generalization for randomized functionalities of the definition from [MPR09].

2.2.2 Some Functionality Classes

In this section we will look at some common and frequently occurring classes of functions in this work. Most of these functionalities are two-party symmetric deterministic function evaluation, i.e. when parties have inputs x and y, both parties receive output $f(x, y)$. We will refer to symmetric secure function evaluation as SSFE. Such functions are represented as matrices where the (i, j)-th entry of the matrix represents the output $f(i, j)$.

Passive Trivial or Decomposable. We emphasize that henceforth we shall always assume that no output renaming is possible for the function being considered. The class of two-party deterministic SSFE which can be securely realized against semi-honest adversaries was characterized independently by Kushilevitz [Kus89] and Beaver [Bea89]. These functions were called decomposable functions due to the combinatorial characterization of their function matrices. Suppose f is a function from $X \times Y$ to Z. If f is a constant function, then f is decomposable. If the input set X can be partitioned into sets $X_1, X_2, \ldots, X_k$ such that f restricted to $X_u \times Y$ and $X_v \times Y$, where $u \neq v$ and $u, v \in [k]$, have no identical output, then we say that f accepts a X-partition. The resulting functions f_u which are restrictions of f to $X_u \times Y$, where $u \in [k]$, respectively are known as sub-functions corresponding to the partition. Similarly, we can define when f accepts a Y-partition. A function f is decomposable, if it accepts a X or Y partition and all sub-functions corresponding to the partition are also decomposable.

We shall only consider the notion of decomposition where where the input space of f is partitioned into maximum possible k partitions. The canonical protocol of a decomposable function is defined as follows. If the function is a constant function, then both parties are aware of the output. Otherwise, if f accepts a X-partition, Alice announces the partition index u such that her input x lies in X_u. If f accepts a Y-partition then Bob announces the partition index u such that his input $y \in Y_u$. After the first message is sent, both parties recursively evaluate the function f_u. Observe that $\mathcal{F}_{\text{CC}}$ has unique decomposition and $\mathcal{F}_{\text{EXCH}}$ has two possible partitions. A function has unique decomposition, if there exists a unique canonical protocol to semi-honest securely evaluate it. The depth of the decomposition refers to the number of rounds in the canonical protocol when maximal partitioning of input spaces is considered.

Passive Complete. This class of functions was identified by [Kil91] who showed that if a function has an embedded OR minor, then it is complete. A two-party SSFE has an embedded OR minor, if there exists inputs $x_0, x_1 \in X$ and $y_0, y_1 \in Y$ and outputs $z_0, z_1 \in Z$ such that $f(x_a, y_b) = z_{a \vee b}$.

Intermediate Functions. When we consider an output space of size at least 4 [CK89, Kre11], there are functions which are neither decomposable nor has an OR minor, for example $\boxed{\begin{smallmatrix}1&1&3&4\\3&2&2&4\\3&4&1&1\end{smallmatrix}}$. Such functions are called intermediate functions.

Exchange-like Functionalities. We define $\mathcal{F}_{\text{EXCH}}^{i\times j} : \mathbb{Z}_i \times \mathbb{Z}_j \rightarrow \mathbb{Z}_{i+j}$ as the function which outputs the sum of the two inputs provided by the two parties. Any function which can be UC securely realized in some $\mathcal{F}_{\text{EXCH}}^{i\times j}$-hybrid against adversaries with unbounded computational power is an exchange like functionality.

Exchange-free Functionalities. A function is exchange free, if it cannot be used to UC-securely realize any form of simultaneous exchange. In particular, $\mathcal{F}$ is exchange free if $\mathcal{F}_{\text{EXCH}}^{2\times 2}$ does not have a UC secure protocol in the $\mathcal{F}$-hybrid against adversaries with unbounded computational power. A useful combinatorial property of exchange-free functionalities which are passive trivial is that they have unique decomposition. We emphasize that it is not necessary that any uniquely decomposable function need not be exchange free, for example $\boxed{\begin{smallmatrix}1&1&2\\5&6&2\\4&3&3\end{smallmatrix}}$.

Trivial or UC-trivial functionalities. These functionalities evaluate extremely simple functions which depend on the input of only one party. These functions are of the form $f(x, y) = h(x)$ or $h(y)$ and have UC-secure protocols even against adversaries with unbounded computation power [CKL03, Lin04, PR08].

Some Example Deterministic Functions. We mention some common deterministic functions which frequently appear in this study.

- $\mathcal{F}_{\text{CC}}^{i,j}$: Alice and Bob have input space $2\mathbb{Z}_i$ and $2\mathbb{Z}_j + 1$ respectively. The functionality announces the higher of the two inputs provided by the two parties. Each of these functionalities are uniquely decomposable and are standalone trivial.

- $\mathcal{F}_{\text{EXCH}}^{i,j}$: Alice and Bob have input space $\mathbb{Z}_i$ and $\mathbb{Z}_j$ respectively. The functionality announced the sum of inputs provided by the two parties, which is an element in $\mathbb{Z}_{i+j}$. Although, these functionalities are decomposable, they do not have unique decomposition and are also not standalone trivial.

- Intermediate functions: The function represented by the following matrix is semi-honest non-trivial and incomplete as well.

$$\begin{pmatrix}1 & 1 & 2\\ 4 & 5 & 2\\ 4 & 3 & 3\end{pmatrix}$$

Randomized functionalities. A randomized SFE functionality is specified by functions $f_1, f_2 : X \times Y \times R \to Z$. The functionality takes inputs $x \in X$ from Alice, $y \in Y$ from Bob, uniformly samples $r \in R$ and outputs $f_1(x, y, r)$ and $f_2(x, y, r)$ to Alice and Bob, respectively. An important example is the common randomness functionality, denoted by $\mathcal{F}_{\text{COIN}}$ (with $X = Y = \{0\}$, $R = \{0, 1\}$, and $f_1(x, y, r) = f_2(x, y, r) = r$). Note that for a given pair of inputs, the outputs to Alice and Bob could be correlated as the same value r is used in both. Two-party symmetric randomized SSFE $\mathcal{F}$ is represented similar to the deterministic ones. The (i, j)-th matrix entry represents the distribution according to which the output is drawn when Alice and Bob have inputs i and j respectively. The distribution is represented as a vector, where the ℓ-th component represents the probability of the ℓ-th output symbol.

We identify two important subclasses of randomized SSFE functionalities:

Selectable sources: One in which one party's input does not affect the output. That is, functions which can be written as $f(x, y, r) = h(x, r)$ for some function h. Note that for different values of x, the function's output distribution may be arbitrary.

Next, we will define the notion of *redundant input*. Suppose only Alice has influence in a selectable source $\mathcal{F}$ implementing the function f. If there exists Alice input x such that the distribution $f(x)$ can be written as linear combination of other distributions, then we say that Alice's input x is redundant. We only consider functions where all redundant inputs have been removed.

Publicly-selectable sources: Those functions which can be written as $f(x, y, r) = (g(x), h(g(x), r))$, for some functions g and h. In this case, the function's output distribution for different values of x must be either identical (when $g(x) = g(x')$) or have disjoint supports (when $g(x) \neq g(x')$, which is included in the function's output). Intuitively, the function's output determines the identity of the random distribution $h(g(x), \cdot)$ that was used.

In these two classes of functionalities, only one party can influence the output, so we say they have *uni-directional influence*. If there exists inputs x, x', x'' for Alice and y, y', y'' for Bob so that $f(x, y') \not\equiv f(x, y'')$, and $f(x', y) \not\equiv f(x'', y)$, then both parties can potentially influence the output, and we say that the functionality has *bi-directional influence*.

Some Examples of Randomized Functions. In Table 2.1, we mention some of the common randomized functions which will be useful to build intuition. The most interesting example is the function which shows how redundancies can be deceptive. The last function in the table seems like an oblivious transfer functionality, but it is not. Because the second input is redundant and, upon removal of that row from

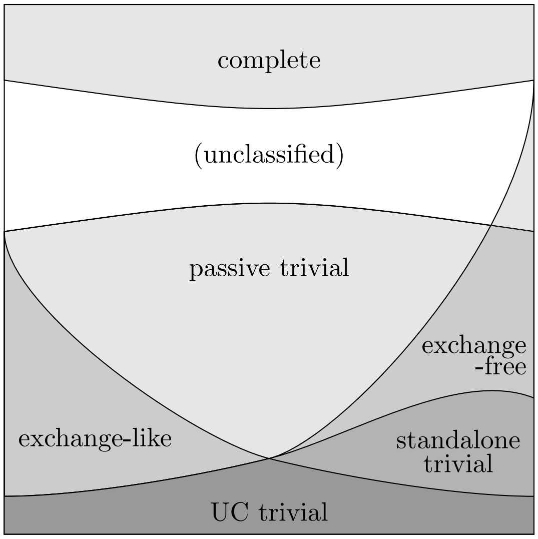

Figure 2.1: A map of various cryptographic complexity classes (of 2-party SSFE functionalities)

the function, the resultant function is a publicly-selectable source implementing a secure communication channel.

Oblivious sampling	$\begin{pmatrix}\langle 1/2, 1/2\rangle \\ \langle 1, 0\rangle\end{pmatrix}$
Publicly-selectable source	$\begin{pmatrix}\langle 1/2, 1/2, 0, 0\rangle \\ \langle 0, 0, 1/2, 1/2\rangle\end{pmatrix}$
$\mathcal{F}_{\textsc{coin}}$ (Function with no influence)	$(\langle 1/2, 1/2\rangle)$
Redundancies	$\begin{pmatrix}\langle 1, 0\rangle \\ \langle 1/2, 1/2\rangle \\ \langle 0, 1\rangle\end{pmatrix}$

Table 2.1: Randomized function examples.

2.2.3 Frontier Analysis

In this section, we present a formal presentation of the transcript generation process.

Protocols and transcript trees. We view a 2-party protocol as a weighted tree of possible transcripts. The leaves of the tree correspond to completed transcripts, on which both parties give output. The tree's internal nodes alternate between "Alice" and "Bob" nodes, corresponding to points in the protocol (identified by partial transcripts) at which Alice and Bob send messages, respectively. Given a party's private input and the transcript so far (i.e., a node in the tree), the protocol assigns probabilities to the outgoing edges (i.e., possible next messages). In some settings we also consider nodes corresponding to invocations of ideal functionalities (like $\mathcal{F}_{\text{COIN}}$), when appropriate. For these the protocol tree assigns probabilities to the outputs of the functionality (the corresponding "messages" included in the transcripts for these steps) according to the probabilities of parties' inputs and the functionality's internal randomness. An execution of the protocol corresponds to a traversal from root to leaf in the tree.

Probabilities and frontiers. We write $\Pr[v|x,y]$ for the probability that the protocol visits node v (equivalently, generates a transcript with v as a prefix) when executed honestly on inputs x and y. Suppose $\pi_A(x, vb)$ is the probability that when Alice executes the protocol honestly with input x and the transcript so far is v, her next message is b. Similarly, we define a probability π_B for Bob. Then (assuming Alice speaks first in the protocol):

$$\Pr[v|x,y] = \pi_A(x, v_1)\pi_B(y, v_1v_2)\cdots = \left[\prod_{i \text{ odd}} \pi_A(x, v_1 \cdots v_i)\right]\left[\prod_{i \text{ even}} \pi_B(y, v_1 \cdots v_i)\right]$$

If we define $\alpha(v,x)$ and $\beta(v,y)$ to be the two parenthesized quantities (equivalently, the product of weights from Alice nodes and Bob nodes in the transcript tree, respectively), then we have $\Pr[v|x,y] = \alpha(v,x)\beta(v,y)$. Thus, in a plain protocol, the two parties make independent contributions to the probability of each transcript. In fact, even if the protocol is allowed to use a selectable source, this property still holds (see Section 3.3). This property of protocols is crucially used in all frontier analysis in this work.

When S is a set of independent nodes in the transcript tree (prefix-free partial transcripts), we define $\Pr[S|x,y] = \sum_{v \in S} \Pr[v|x,y]$, as all the probabilities in the summation are for mutually exclusive events. If $\Pr[F|x,y] = 1$, then we call F a *frontier.* Equivalently, a frontier is a maximal independent set in the transcript tree. In general, a frontier represents a point in the protocol where a certain event happens, usually defined in terms of the probabilities α and β.

2.2.4 Computational Intractability Assumptions

In this section we will look at the definition of some popular computational intractability assumptions. We will define an experiment and define what computation is tough for a computationally bounded adversary.

One-way Functions. Consider an ensemble of functions $f_n : \{0,1\}^n \to \{0,1\}^{m(n)}$, where $m(\cdot)$ is a polynomial. The experiment, draws a random $x \in \{0,1\}^n$ and provides the adversary with n and $y = f_n(x)$.

Next, the adversary outputs a string $x' \in \{0,1\}^n$. If $f(x') = y$ then we say that the adversary wins the challenge. We say that f is a one-way function, if the probability of any efficient adversary winning the challenge is negligible in n. If the function f_n is also a permutation, then f is said to be a one-way permutation. If one-way functions exist, then we say that OWF assumption is true.

Public-key Encryption. A public-key encryption scheme is defined by three algorithms: Gen, Enc and Dec. The key-generation algorithm Gen, takes as input a random tape r and outputs a tuple of private and public key (sk, pk). The encryption algorithm Enc takes as input the public key pk and a message m and outputs a cipher text c. The decryption algorithm Dec takes as input the secret key sk and the cipher text c and outputs its corresponding message m. A correct public-key encryption scheme ensures that $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m$, where $(sk, pk) = \mathsf{Gen}(r)$. In the indistinguishability experiment, we first sample a random tape r and generate a secret and public key pair $(sk, pk) = \mathsf{Gen}(r)$. We send pk to the adversary and receive two challenge messages m_0 and m_1. We choose a random bit b and send the adversary encryption of m_b, i.e. $c = \mathsf{Enc}(pk, m_b)$. Finally, the adversary provides a bit b' representing the fact that it thinks that c is an encryption of $m_{b'}$. The adversary wins the game if $b = b'$. A public-key encryption scheme is secure, if the advantage of an efficient algorithm winning the game is at most negligible. If there exists a public-key encryption protocol then we say that PKE assumption is true.

Key-agreement Protocol. In key-agreement two parties Alice and Bob are interested in agreeing on a secret key which is not leaked to an eavesdropping adversary. Consider a two party protocol between Alice and Bob that runs for r rounds. At the end of the protocol, Alice and Bob agree to a secret key s with high probability. Suppose the generated transcript τ is provided to an adversary. If the adversary cannot guess s with non-negligible probability then we say that the protocol is a secure key-agreement protocol. If such protocol exists then (r-round) KA assumption holds. Observe that PKE is equivalent to 2-round KA protocol.

Semi-honest Oblivious Transfer Protocol. An oblivious transfer involves two parties: Sender and Receiver. The sender has two bits x_0 and x_1 and the receiver has a choice bit b. At the end of the protocol the receiver finds out x_b. The security of the sender demands that any adversary using the receiver's view has only negligible advantage in guessing $x_{(1-b)}$, And the security of the receiver requires that any adversary using the sender's view can guess b only with negligible advantage. If both these conditions hold for a protocol, then that protocol is a semi-honest secure protocol for oblivious transfer. Similar to the definition of KA experiment, we can also define $\mathsf{sh\text{-}OT}$ with r-rounds. In this work, we shall not need such a definition but such definitions have been used elsewhere while considering computational assumptions, for example [GKM+00].

Chapter 3

Information Theoretic Irreducibility

In this chapter, we consider a fundamental question: *How cryptographically useful is a trusted source of public coins against adversaries with unbounded computational power?*

While there are several instances in cryptography where a common random string or a trusted source of public coins is very useful (e.g. [BFM88, CLOS02]), we show severe limitations to its usefulness[1] in secure two-party computation, without — and sometimes even with — computational intractability assumptions. In contrast, it is well known that more general correlated private random variables can be extremely powerful [Bea95]. Given that for semi-honest security common randomness is useless (as one of the parties could sample and broadcast it), it is not surprising that it should turn out to be not as powerful as general correlated random variables. However, despite its fundamental nature, the exact power of common randomness has not yet been characterized. We show:

- For two-party secure function evaluation (SFE) of *deterministic* functions, being given a source of common randomness is useless, irrespective of any computational complexity assumptions, when considering security in the standalone setting.[2]
- Clearly a source of common randomness can be useful for realizing *randomized* functionalities. However, in the case of UC security, we show that a source of common coins can be useful only in a trivial sense (unless restricted to the computationally bounded setting, and intractability assumptions are employed). We show that any UC-secure protocol that uses common coins for evaluating a randomized function can be replaced by a protocol of the following simple form: one of the parties announces a probability distribution, based deterministically on its input, and then the two parties sample an outcome from this distribution using freshly sampled common coins. We call the resulting functionality a *publicly-selectable source.*

[1] We say that a source of common randomness is useless in realizing some 2-party functionality $\mathcal{F}$ if either $\mathcal{F}$ could be realized without using the given source or $\mathcal{F}$ cannot be realized even given the source. Note that we consider only the feasibility question and not any efficiency issues.

[2] In the case of UC security, it follows from the results in [MPR10a] that a source of common randomness is useless except in Cryptomania, where it is a complete functionality.

These results are actually proven for a class of sources more general than coin tossing, namely *selectable sources* – that let one of the parties (secretly) specify which among a set of distributions should be used by the source. We highlight two aspects of these results:

Non-blackbox analysis of protocols. In deriving the impossibility results our analysis crucially relies on the communication and information structure of protocols. We build on the "frontier analysis" paradigm in [CI93, MPR09, MPR10a], but significantly extend its power, among other things, to enable analyzing protocols for arbitrary randomized functionalities, and protocols using randomized functionalities.

These results (and hence proofs) are necessarily of a *non-relativizing* nature — if the protocol has access to another trusted functionality (more sophisticated than common randomness), the impossibility results no longer hold. Specifics about the common randomness functionality are (and must be) used in our proofs. Such low-level analysis of protocols, we believe, is crucial to understanding the power and complexity of multi-party computation primitives.

Understanding randomized functionalities. Secure evaluation of *randomized* functions has in general been a poorly understood area. In particular, to date it remains open to characterize which randomized functions can be securely realized even against computationally unbounded passive (honest-but-curious) adversaries — a problem that was solved for deterministic functions twenty years ago [Bea89, Kus89]. Much of the study of randomized functionalities has been focused on in-depth understanding of the simplest such functionality — namely generating shared fair coins (e.g., see [Cle86, IL89, CI93, MNS09] and references therein). Our results provide significant insight into *other* randomized functionalities as well, and their connections to computational intractability assumptions. In particular, our results involve two interesting classes of randomized functionalities, namely *selectable sources* and *publicly-selectable sources*.

3.1 Overview

Frontier analysis. The bulk of our results take the form of statements of cryptographic impossibility. That is, we show that a protocol for a given cryptographic task is impossible (or else implies a certain computational primitive like one-way functions). Such impossibility results have been a core challenge in cryptography. In this chapter, we present a powerful battery of techniques that we use to analyze 2-party protocols, which we broadly call "frontier analysis."

The basic outline of a frontier analysis is as follows. We first interpret a protocol as a tree of possible transcripts, with weights corresponding to the probability that the protocol assigns to each message, based on the parties' inputs. Within this tree, we identify "frontiers", which are simply a collection of nodes (partial transcripts) that form a cut and an independent set. Intuitively, these frontiers correspond to points in the protocol when some condition is satisfied for the first time, where the condition in question depends on the kind of analysis needed: for example, the first place the transcript leaks "significant" information about a party's input, or the first place that common coins have made a "significant" influence on the protocol's output.

Impossibility proofs using frontier analysis proceed by showing that frontiers of certain kind exist, often showing that multiple frontiers must be encountered in a specific order, and then showing that an adversary can effect an attack by exploiting the properties of these frontiers.

Common coins are not useful in SFE protocols. We show that against computationally unbounded adversaries (more precisely, against adversaries that can break one-way functions), any 2-party deterministic SFE (in which both parties receive the same output) functionality that can be securely realized given a trusted coin-tossing functionality can in fact be securely realized without it. This is most interesting for the standalone setting, because if one-way functions do exist then a standalone-secure coin-tossing protocols exist, so again access to a trusted coin-tossing functionality is redundant.[3]

We start off by showing that there is no secure protocol for evaluating boolean XOR given a coin-tossing functionality. In many ways these functionalities have similar "complexity" (in particular, neither is complete, and both are trivial to realize against passive adversaries), so establishing a qualitative separation between them is interesting in itself. In a protocol for XOR, either party may be the first to reveal information about their input, and the two parties can even gradually reveal more and more information about their input in an interleaved fashion. We define a frontier corresponding to the first point at which some party has revealed "significant" information about its input. Then we define an attack that can be carried out when the protocol crosses this frontier. Since a large class of SFE functionalities can be used to securely realize XOR, the impossibility extends to these functionalities as well.

We then use the combinatorial characterizations of Symmetric Secure Function Evaluation (SSFE) functionalities (obtained using frontier analysis) from [MPR09] to extend the result to arbitrary SSFE functionalities (instead of just XOR). Further, using an extension of a result in [Kil00], we extend this to arbitrary SFE functionalities by associating a symmetric SFE with every general SFE that has a secure protocol using a source of common randomness.

[3] A recent result in [MPR10a] gives a sharp result for the case of UC security: the coin-tossing functionality is useful in realizing further deterministic SFE functionalities if and only if there exists a semi-honest oblivious transfer protocol. However neither the result nor the approach in [MPR10a] extends to the standalone setting. Also, our result is applicable to not just symmetric functionalities and coin-tossing, but extends to general SFE functionalities and all selectable sources.

For randomized SFE, common coins help only in a trivial sense. We show that common coins are useful in constructing UC-secure protocols for randomized SFE functionalities only for the class of publicly-selectable sources (Theorem 2). For this result, we exploit the versatility of the frontier analysis and also employ a geometric analysis of the space of effective probability distributions.

The frontier analysis is carried out for an SSFE functionality, and then the result is extended to general SFE functionality separately. For a randomized SSFE functionality, for each pair of inputs, the output is specified by a distribution (over a finite output alphabet). This distribution can be represented as a vector in d-dimensional real space where d is the size of the output alphabet. By considering all possible inputs, we obtain a set of points in this space as legitimate output distributions. But since the parties can choose their input according to any distribution they wish, the entire convex hull of these points is the set of legitimate output distributions. Note that the vertices of this polytope correspond to the output distributions for various specific input choices.

In analyzing a protocol for such a functionality, we define *two* very different frontiers: one intuitively captures the last point in the protocol where the parties' inputs have any noticeable influence over the output distribution. The other intuitively captures the first point where the common coins have had a non-trivial influence on the output distribution.

Defining these frontiers is a delicate task, but once they are defined, we can show that, for the protocol to be UC-secure, the two frontiers must be encountered in the order listed above. Thus there is always a point within the protocol where the parties' inputs have stopped influencing the output, yet the public coins have not yet started influencing the output in a non-trivial way. At this point, we can show that the output distribution is uniquely determined, and that the subsequent coins are simply used to sample from this publicly-chosen distribution.

Then, on each node in the first frontier the conditional output distribution is still within the polytope. On the other hand, since the input influence has ceased at this point, for any fixed input, its output distribution must be determined by this frontier: i.e., it must be a convex combination of the conditional output distributions at the nodes on the frontier. That is, the output distribution for this input is a convex combination of conditional output distributions which are all themselves within the polytope. Now, (without loss of generality, as it turns out) we can consider inputs whose output distributions are vertices of the polytope. Then, *for all nodes in the frontier* the conditional output distribution must coincide with the final distribution itself. Thus on reaching this frontier in the protocol, the output distribution is revealed (as a deterministic function of the inputs) and the rest of the protocol simply samples from this distribution.

Finally, we extend this result also to general SFE (instead of just symmetric SFE) functionalities, in the same way as for deterministic functionalities.

Selectable sources. Selectable sources are an interesting class of randomized functionalities with an intermediate level of complexity: they can be more complex than a (fixed) source of common randomness, yet they are simple enough that we can show that they are as useless as common randomness when it comes to securely realizing deterministic functionalities. The extension is observed by following the analysis for the case of the source of common randomness, and identifying the properties that it relies on. We do not know at this point whether these are exactly all the functionalities which are useless for realizing SFE functionalities, but based on our understanding so far, we conjecture that they are.

Related Results. Frontier analysis is possibly implicit in previous works on proving impossibility or lower bounds for protocols. For instance, the analysis in [CI93] very well fits our notion of what frontier analysis is. The analysis of protocols in [CK89, Bea89, Kus89] also have some elements of a frontier analysis, but of a rudimentary form which was sufficient for analysis of perfect security. In [MPR09] frontier analysis was explicitly introduced and used to prove several protocol impossibility results and characterizations. [KMR09] also presented similar results and used somewhat similar techniques (but relied on analyzing the protocol by rounds, instead of frontiers, and suffered limitations on the round complexity of the protocols for which the impossibility could be shown).

3.2 Handling General SFE Functionalities

Frontier analysis is most naturally applied to protocols realizing SSFE functionalities — that is, functionalities which give the same output to both parties. So we derive our results for such functionalities. However, we can then extend our characterizations to apply to SFE functionalities with unrestricted outputs using the following lemma:

Lemma 2. *Suppose $\mathcal{H}$ is a functionality that has a passive-secure protocol in the plain model. If $\mathcal{H}$ is useful in UC- or standalone-securely realizing a (possibly randomized) SFE functionality $\mathcal{F}$, then there exists a* **symmetric** *SFE functionality $\mathcal{F}^*$ such that $\mathcal{F}^*$ is isomorphic to $\mathcal{F}$, and $\mathcal{H}$ is useful in (respectively, UC- or standalone-) securely realizing $\mathcal{F}^*$.*

Here, being useful or not is in the sense of the definition given in Section 2.2.1.

Proving Lemma 2 essentially involves relating SSFE and SFE functionalities. As it turns out, relating symmetric and unrestricted functionalities is most convenient in the setting of passive security. In that setting, we associate with every SFE functionality $\mathcal{F}$ a symmetric functionality which is simply the maximal "common information" provided to the two parties by $\mathcal{F}$. (See proof of Lemma 4 for a combinatorial description of this function.) Following [Kil00] it is not hard to show that if an SFE functionality $\mathcal{G}$ is not isomorphic to its (symmetric-output) common information functionality then $\mathcal{G}$ must be complete in the passive security setting.

To apply this result, however, we must be careful in relating passive security and active security. It is not necessarily the case that an actively secure protocol implies a passively secure protocol (since in the passive security setting, the security reduction must map passively corrupt adversaries to passively corrupt simulators). In Lemma 3 we show that every SFE functionality is isomorphic to a functionality that is "deviation-revealing" [PR08]. Such functionalities have the property that active-secure protocols imply passive-secure protocols. Using these two results, we are able to transition from active to passive security, and then argue about generalized vs. symmetric output.

Proof. We say that two SFE functionalities $\mathcal{F}$ and $\mathcal{G}$ are *isomorphic* if there is a *local* protocol for UC-securely realizing $\mathcal{F}$ in the $\mathcal{G}$-hybrid model, and vice-versa. By *local*, we mean that the protocol (say, the protocol for $\mathcal{F}$ in the $\mathcal{G}$-hybrid model) makes only one call to the ideal functionality $\mathcal{G}$ and performs no other communication. Local protocols allow each party to do no more than locally "translate" both the input from the environment and the output from $\mathcal{G}$. This translation may be randomized, especially in the case that $\mathcal{F}$ and $\mathcal{G}$ are randomized.

We say that an input for Alice x is *redundant* in an SFE $\mathcal{F}$ if $\mathcal{F}$ is isomorphic to a variant $\mathcal{F}_{-x}$ of $\mathcal{F}$ that does not allow input x from Alice. In other words, the effect of x can be achieved by having Alice locally translate her inputs and outputs to/from $\mathcal{F}$, using only inputs other than x. In this definition of redundancy, the protocol for $\mathcal{F}_{-x}$ in the $\mathcal{F}$-hybrid model is always the dummy protocol; the simulator for corrupt Alice in the $\mathcal{F}$ protocol in the $\mathcal{F}_{-x}$-hybrid model is also the dummy simulation. The simulator for the $\mathcal{F}_{-x}$ protocol and Alice's protocol for $\mathcal{F}$ coincide, and they correspond to Alice's "translation" technique for obviating the input x. Bob's protocol is the dummy protocol without loss of generality.

[PR08] define a property of functionalities called *deviation-revealing*, which relates UC security to passive security. UC security considers only actively corrupt adversaries — as such, it does not require that passively corrupt adversaries (who receive inputs from the environment on which to follow the protocol) are mapped to passively corrupt simulators (i.e., a simulator that runs the dummy protocol with the functionality).

For the purposes of this result, we define deviation-revealing slightly more restrictively than [PR08], requiring a condition for standalone security as well. We say that a functionality $\mathcal{F}$ is *deviation-revealing* if every UC-secure *or standalone-secure* protocol for $\mathcal{F}$ in the $\mathcal{G}$-hybrid model is itself a passive-secure protocol for $\mathcal{F}$ in the $\mathcal{G}$-hybrid model. But if $\mathcal{F}$ is deviation-revealing, then without loss of generality the simulator for a passively corrupt adversary can be passively corrupt. The name "deviation-revealing" comes from the fact that the functionality's behavior would reveal to an environment whether a party is interacting with $\mathcal{F}$ using the dummy protocol or deviating from it.

Lemma 3. *For every SFE functionality $\mathcal{F}$ there is a deviation-revealing functionality $\mathcal{G}$ that is isomorphic to it.*

Proof. Given an SFE $\mathcal{F}$, we define $\mathcal{G}$ by iteratively removing redundant inputs in $\mathcal{F}$ (for both parties). We do not require that removing redundant inputs results in a unique $\mathcal{G}$. Clearly $\mathcal{G}$ and $\mathcal{F}$ are isomorphic, and it suffices to show that $\mathcal{G}$ is deviation-revealing.

Let π be any UC-secure or standalone-secure protocol for $\mathcal{G}$ in the $\mathcal{H}$-hybrid model. We must show that π is itself also passive-secure in the $\mathcal{H}$-hybrid model. Consider a passive adversary $\mathcal{A}$ for π — that is, the adversary receives inputs from the environment and executes π honestly on those inputs, but also outputs its entire view to the environment. Let $\mathcal{S}$ be the simulator for this adversary, and it suffices to show that $\mathcal{S}$ can be made to interact with $\mathcal{G}$ according to the dummy protocol without loss of generality.

Consider a class of environments that inspect only the inputs and outputs of the parties, and in particular ignore $\mathcal{A}$'s reported view of the protocol. By the correctness of π, an interaction with $\mathcal{A}$ is indistinguishable from an interaction with $\mathcal{G}$ in which all parties run the dummy protocol, for this class of environments.

Suppose such an environment gives input x to $\mathcal{S}$, and condition on the event that its simulator $\mathcal{S}$ sends an input other than x to $\mathcal{G}$. This $\mathcal{S}$ is expected to also return the output from $\mathcal{G}$, since the original passive adversary returned the output from π. By the security of π, this interaction is indistinguishable from an interaction with ideal $\mathcal{G}$ in which all parties run the dummy protocol, for this class of environments. Thus $\mathcal{S}$ is effecting a *local protocol* which demonstrates that the input x is *redundant.* Since $\mathcal{G}$ contains no redundant inputs, we conclude that this event (environment provides x but $\mathcal{S}$ sends an input other than x) happens with only negligible probability. Without loss of generality, we can add a wrapper around $\mathcal{S}$ that aborts if $\mathcal{S}$ sends an input other than the one provided by the environment. This wrapped simulator is still a sound simulation and is a passive simulator. □

Lemma 4. *For every SFE functionality $\mathcal{G}$ that has a passive secure protocol in the plain model, there is a symmetric functionality $\mathcal{G}'$ that is isomorphic to it.*

Proof. We define the symmetric functionality $\mathcal{G}'$ to be the "common information" that Alice and Bob get from $\mathcal{G}$. This is best described by representing $\mathcal{G}$ as a bipartite graph G: the set of nodes on the left are (x, a) for each possible input value x for Alice x and output value a for Alice; similarly, the set of nodes on the right are (y, b) for all possible inputs y and outputs b for Bob. There is a weighted edge between (x, a) and (y, b) with weight $\Pr[a, b|x, y]$, namely, the probability that Alice and Bob get outputs a and b when they send x and y as their respective inputs to $\mathcal{G}$. If this weight is 0, then we consider the edge to be absent. $\mathcal{G}'$ is defined as an SSFE functionality which takes x and y from the parties, samples the outcome (a, b) according to $\mathcal{G}$, and returns to both parties the *connected component* H containing the edge $((x, a), (y, b))$ in G. Observe that $\mathcal{G}'$ gives the same output to both parties.

It suffices to show that if $\mathcal{G}'$ and $\mathcal{G}$ are not isomorphic, then $\mathcal{G}$ cannot have a passive secure protocol in the plain model. For this we rely on a result by Kilian [Kil00] to show that in this case $\mathcal{G}$ will actually be *complete* for passive security, and hence cannot have a passive secure protocol in the plain model (unless we impose computational restrictions and assume that there is such a protocol for oblivious transfer).

Due to the restriction of *local* protocols, we see that $\mathcal{G}$ and $\mathcal{G}'$ are isomorphic if and only if, given the connected component H and their respective inputs, Alice and Bob can *independently* sample outcomes that are jointly distributed as outcomes from $\mathcal{G}$. This is possible only when there is a labeling of every vertex $q(x, a)$ (or $q(y, b)$) so that $\Pr[a, b|x, y] = q(x, a)q(y, b) \Pr[H|x, y]$. By $\Pr(H|x, y)$, we mean the probability that $\mathcal{G}'$ outputs H on inputs x and y.

Now suppose no such labeling exists. Then we claim that $\mathcal{G}$ must be *complete* for passive security. We adapt an argument of Kilian, who proved an analogous statement for a special class of (deterministic) "asymmetric" SFEs $\mathcal{G}$ (Theorem 1.3 in [Kil00]).[4]

We consider two cases which exhaustively characterize the condition described above:

Case 1: Suppose there exists $(x_0, a_0), (y_0, b_0), (y_1, b_1)$ such that $\Pr[a_0, b_0|x_0, y_0] > \Pr[a_0, b_1|x_0, y_1] > 0$ (or vice-versa with the roles of Alice and Bob exchanged). Then there must be a value a_1 such that $\Pr[a_1, b_0|x_0, y_0] < \Pr[a_1, b_1|x_0, y_1]$

Then consider the following passive protocol using $\mathcal{G}$, where Bob has input m:

1. Bob chooses a random bit t. The parties evaluate $\mathcal{G}$ twice, on inputs (x_0, y_t) and (x_0, y_{1-t}).
2. If Bob did not receive output sequence (b_1, b_1) or Alice did not receive a sequence of outputs in the set $\{(a_0, a_1), (a_1, a_0), (a_0, a_0)\}$ then the parties repeat step 1.
3. Bob sends $M = m \oplus t$ to Alice. If Alice received (a_0, a_1), she guesses $\hat{t} = 0$; if Alice received (a_1, a_0), she guesses $\hat{t} = 1$; otherwise, she sets $\hat{t}$ randomly. Alice locally outputs $M \oplus \hat{t}$.

The analysis of this protocol closely follows that of [Kil00] (Lemma 5.2). Briefly, Bob's choice t is uniformly distributed conditioned on Alice receiving (a_0, a_0). In this case, she receives no information about Bob's

[4] Kilian does not state the result in terms of isomorphism or common information. But the combinatorial condition is identical to the above.

input m. Otherwise, Alice's guess of $\hat{t}$ is biased towards Bob's choice of t and she learns partial information about m. The protocol therefore gives a "noisy" variant of Rabin OT that can be refined using the techniques described in [Kil00].

Case 2: Suppose Case 1 does not hold and that there exist $(x_0, a_0), (x_1, a_1), (y_0, b_0), (y_1, b_1)$ such that $\Pr[a_0, b_0|x_0, y_0] = 0$, yet each of $\Pr[a_0, b_1|x_0, y_1], \Pr[a_1, b_0|x_1, y_0], \Pr[a_1, b_1|x_1, y_1]$ are nonzero. Since Case 1 does not hold, then these latter three probabilities must in fact be equal. Then consider the following passive protocol using $\mathcal{G}$:

1. Alice chooses random bit s. Bob chooses random bit t. Alice sends x_s to $\mathcal{G}$ and Bob sends y_t to $\mathcal{G}$.

2. If Alice did not receive output a_s or Bob did not receive output b_t, then the parties repeat step 1.

3. Alice locally outputs s. Bob locally outputs t.

This protocol allows Alice and Bob to generate correlated pairs (s, t) that are *uniformly* distributed in $\{(0, 1), (1, 0), (1, 1)\}$. Using the techniques spelled out in [Kil00], such correlated pairs can be used to implement a passively secure OT. □

We can now prove Lemma 2:

Lemma 2 (restated). *Suppose $\mathcal{H}$ is a functionality that has a passive-secure protocol in the plain model. If $\mathcal{H}$ is useful in UC- or standalone-securely realizing a (possibly randomized) SFE functionality $\mathcal{F}$, then there exists a* **symmetric** *SFE functionality $\mathcal{F}^*$ such that $\mathcal{F}^*$ is isomorphic to $\mathcal{F}$, and $\mathcal{H}$ is useful in (respectively, UC- or standalone-) securely realizing $\mathcal{F}^*$.*

Proof. First note that if $\mathcal{F}$ is isomorphic to $\mathcal{F}^*$, then $\mathcal{H}$ is useful in securely realizing $\mathcal{F}$ if and only if $\mathcal{H}$ is useful in securely realizing $\mathcal{F}^*$. (This is because, if there is a protocol for $\mathcal{F}$ in the $\mathcal{H}$-hybrid model there is one for $\mathcal{F}^*$, and if there is no protocol for $\mathcal{F}$ in the plain model, there is none for $\mathcal{F}^*$ either.) So it is enough to give an SSFE functionality that is isomorphic to $\mathcal{F}$.

If $\mathcal{H}$ is useful in UC-/standalone-securely realizing a randomized SFE functionality $\mathcal{F}$, then $\mathcal{F}$ has a (respectively, UC- or standalone-) secure protocol in the $\mathcal{H}$-hybrid model. Let $\mathcal{G}$ be the deviation-revealing functionality guaranteed by Lemma 3. Because $\mathcal{G}$ is isomorphic to $\mathcal{F}$, we have that $\mathcal{G}$ has a (respectively UC- or standalone-) secure protocol in the $\mathcal{H}$-hybrid protocol. Then, since $\mathcal{G}$ is deviation-revealing, the same protocol is also passively secure in the $\mathcal{H}$-hybrid model. By our assumption, $\mathcal{H}$ has a passive-secure protocol in the plain model; so by composing these two protocols we can obtain a passive secure protocol for $\mathcal{G}$ in the plain model. Now, by Lemma 4, there is an SSFE functionality $\mathcal{F}^*$ that is isomorphic to $\mathcal{G}$. Thus $\mathcal{F}^*$ is our desired SSFE that is isomorphic to $\mathcal{F}$. □

3.3 Selectable Sources are Useless for Deterministic SFE

In this section we will show that any selectable source is useless for securely realizing any deterministic SFE functionality against computationally unbounded adversaries. In particular this shows that $\mathcal{F}_{\text{COIN}}$ is useless for realizing any deterministic SFE functionality.

Theorem 1. *Suppose $\mathcal{F}$ is a 2-party deterministic SFE and $\mathcal{G}$ is a selectable source. Then $\mathcal{F}$ has a standalone-secure (resp. UC-secure) protocol in the $\mathcal{G}$-hybrid model against computationally unbounded adversaries if and only if $\mathcal{F}$ has a standalone-secure (resp. UC-secure) protocol in the plain model.*

To give an overview of our techniques, we present the result for the special case of $\mathcal{F} = \mathcal{F}_{\text{EXCH}}$ and $\mathcal{G} = \mathcal{F}_{\text{COIN}}$. Then we describe the modifications necessary to consider arbitrary $\mathcal{F}$ and arbitrary selectable source $\mathcal{G}$, respectively.

The case of $\mathcal{F}_{\text{EXCH}}$ and $\mathcal{F}_{\text{COIN}}$. This special case illustrates our new frontier-based attack. It is well-known that there is no standalone-secure (or UC-secure) protocol for $\mathcal{F}_{\text{EXCH}}$ in the plain model (cf. the complete characterization of [KMR09, MPR09]). Also note that standalone security is a special case of UC security. Thus it suffices to show the following:

Lemma 5. *There is no standalone-secure protocol for $\mathcal{F}_{\text{EXCH}}$ using $\mathcal{F}_{\text{COIN}}$, against computationally unbounded adversaries.*

Proof. Before we start on this result, we note that this result also rules out UC secure protocol of $\mathcal{F}_{\text{EXCH}}$ in the $\mathcal{F}_{\text{COIN}}$-hybrid when adversaries have unbounded computational power. But that result can be directly proven using the approach in Chapter 4 where we shall show that $\mathcal{F}_{\text{EXCH}} \sqsubseteq^{\text{PPT}} \mathcal{F}_{\text{COIN}}$ implies that the $\mathsf{sh\text{-}OT}$ assumption is true, which cannot be the case against adversaries with unbounded computational power. Thus, we emphasize that this result highlights that even standalone reduction of $\mathcal{F}_{\text{EXCH}}$ to $\mathcal{F}_{\text{COIN}}$ is not possible in the statistical setting.

Suppose for contradiction π is a standalone-secure protocol for $\mathcal{F}_{\text{EXCH}}$ in the $\mathcal{F}_{\text{COIN}}$-hybrid model. Recall that in $\mathcal{F}_{\text{EXCH}}$, Alice chooses an input $x \in \{0,1\}$, Bob chooses an input $y \in \{0,1\}$, and both parties learn $x \oplus y$. We will show an attack against π that violates the security guarantee of $\mathcal{F}_{\text{EXCH}}$— specifically, we will show an attack whereby the honest party chooses its input at random, yet its output is significantly biased.

This is indeed a violation of security since in the ideal world the corresponding output must be an unbiased bit.

Without loss of generality, assume that every other round of the protocol is an access to $\mathcal{F}_{\text{COIN}}$. We only use the property that the probability of each transcript consists of independent probability contributions from Alice, Bob, and $\mathcal{F}_{\text{COIN}}$. Let $\varepsilon = \varepsilon(k)$ denote the security error (maximum deviation between ideal world and real world) of the protocol, thus ε is negligible in the security parameter k.

Define α and β as in Chapter 2, and let γ be the probability contribution from $\mathcal{F}_{\text{COIN}}$. Thus, for every partial transcript v we can express $\Pr[v|x,y] = \alpha(v,x)\beta(v,y)\gamma(v)$. Now, for every partial transcript v, define

$$\delta_A(v) = \frac{|\alpha(v,0) - \alpha(v,1)|}{\alpha(v,0) + \alpha(v,1)} \qquad \text{and} \qquad \delta_B(v) = \frac{|\beta(v,0) - \beta(v,1)|}{\beta(v,0) + \beta(v,1)}.$$

δ_A and δ_B are well-defined after we exclude any nodes that have $\alpha(v,0) = \alpha(v,1) = 0$ or $\beta(v,0) = \beta(v,1) = 0$. Intuitively, $\delta_A(v)$ and $\delta_B(v)$ measure how much Alice's or Bob's input affects the probability of reaching v, respectively. For instance, $\delta_A(v) = 0$ means that the partial transcript v contains no information about Alice's input (in fact, it is distributed independent of her input); $\delta_A(v) = 1$ means that the partial transcript v completely reveals Alice's input — it is uniquely determined by v.

Let $0 < \mu \leq 1$ be a fixed parameter to be defined later, and define the following sets:

$$F_A = \{v \mid \delta_A(v) \geq \mu \text{ and no proper prefix of } v \text{ satisfies } (\delta_A(v) \geq \mu \text{ or } \delta_B(v) \geq \mu)\}$$

$$F_B = \{v \mid \delta_B(v) \geq \mu \text{ and no proper prefix of } v \text{ satisfies } (\delta_A(v) \geq \mu \text{ or } \delta_B(v) \geq \mu)\}$$

$$F_C = \{v \mid v \text{ is a complete transcript and no proper prefix of } v \text{ satisfies } (\delta_A(v) \geq \mu \text{ or } \delta_B(v) \geq \mu)\}$$

It is easy to see that $F_A \cup F_B \cup F_C$ indeed constitute a complete frontier. Intuitively, F_A and F_B represent the first place where Alice or Bob has revealed "significant" information about their input, respectively, where the parameter μ measures the amount of significance. F_C represents the remaining transcripts needed to extend $F_A \cup F_B$ to a frontier.

First, we argue that F_C is only reached with negligible probability during honest executions of the protocol. Intuitively, the transcript must eventually reveal both parties inputs, since the transcript contains at least the output $x \oplus y$ and any two of $\{x, y, x \oplus y\}$ uniquely determine the third quantity. The following proposition is useful:

Proposition 1. *If* $|p - q|/(p + q) < c$, *then* $\frac{p}{q}, \frac{q}{p} \in (\frac{1-c}{1+c}, \frac{1+c}{1-c})$.

Thus, for any $v \in F_C$, we have $\alpha(v,0)/\alpha(v,1), \beta(v,0)/\beta(v,1) \in (\frac{1-\mu}{1+\mu}, \frac{1+\mu}{1-\mu})$. Since transcripts in F_C are complete transcripts, each one uniquely determines the output of the parties. Partition F_C into $F_C^{(0)}$ and $F_C^{(1)}$, where $F_C^{(b)}$ are the transcripts on which Alice outputs b. Note that by the correctness of the protocol, $\Pr[F_C^{(x \oplus y \oplus 1)} | x, y] \leq \varepsilon$. Then

$$\begin{aligned}\Pr[F_C \mid x=0, y=0] &= \sum_{v \in F_C^{(1)}} \Pr[v | x=0, y=0] + \sum_{v \in F_C^{(0)}} \Pr[v | x=0, y=0] \\ &\leq \varepsilon + \frac{1+\mu}{1-\mu} \sum_{v \in F_C^{(0)}} \Pr[v | x=1, y=0] \leq \frac{1+\mu}{1-\mu}\varepsilon + \varepsilon = \frac{2\varepsilon}{1-\mu} = \mathrm{negl}(k).\end{aligned}$$

Here we assume that μ is a constant independent of k. Similarly, $\Pr[F_C | x, y] \leq \mathrm{negl}(k)$ for all $x, y \in \{0, 1\}$.

Now partition F_A and F_B respectively into the following sets:

$$\begin{aligned} F_{A0} &= \{v \in F_A \mid \alpha(v,0) > \alpha(v,1)\} & \qquad F_{B0} &= \{v \in F_B \mid \beta(v,0) > \beta(v,1)\} \\ F_{A1} &= \{v \in F_A \mid \alpha(v,1) > \alpha(v,0)\} & \qquad F_{B1} &= \{v \in F_B \mid \beta(v,1) > \beta(v,0)\} \end{aligned}$$

Then $F_{A,x}$ is the point in the protocol at which the transcript is significantly biased towards Alice having input x; similarly for $F_{B,y}$.

By symmetry, suppose $\Pr[F_{B0} \mid x=0, y=0]$ is the maximum of the four values

$$\Big\{ \Pr[F_{A0} \mid x=0, y=0], \Pr[F_{A1} \mid x=0, y=0], \Pr[F_{B0} \mid x=0, y=0], \Pr[F_{B1} \mid x=0, y=0] \Big\}.$$

Then, since $\Pr[F_C \mid x=0, y=0] < \mathrm{negl}(k)$, we have $\Pr[F_{B0} \mid x=0, y=0] \geq \frac{1}{4} - \mathrm{negl}(k)$.

We now construct a strategy for a corrupt Alice that will bias Bob's output towards 1 when Bob is executing π on a randomly chosen bit y. Alice's strategy is to run the protocol honestly with input $x = 0$, until the transcript reaches a node v on frontier F. If $v \notin F_{B0}$, then she continues the execution honestly. Otherwise (i.e., $v \in F_{B0}$) she switches her input to 1 (by sampling a state consistent with the current transcript and the input 1) and continues the execution honestly with her new state.

Let $\textsc{out}$ denote the output of Bob in the protocol, and let p' denote the probability in the interaction described above (honest Bob choosing a random input y, and Alice running the strategy described). It suffices to show that $\left|p'[\textsc{out} = 0] - \frac{1}{2}\right|$ is bounded by a positive constant.

We split the analysis into cases. Let F_{B0} denote the event that the transcript intersects the frontier F

at a point in F_{B0}. Then

$$\begin{aligned} p'[\textsc{out} = 0] = \frac{1}{2}\Big[& p'[\textsc{out} = 0 \wedge F_{B0} \mid y = 0] + p'[\textsc{out} = 0 \wedge \overline{F_{B0}} \mid y = 0] \\ & + p'[\textsc{out} = 0 \wedge F_{B0} \mid y = 1] + p'[\textsc{out} = 0 \wedge \overline{F_{B0}} \mid y = 1]\Big] \end{aligned}$$

We bound each of these four quantities separately.

First, $p'[\textsc{out} = 0 \wedge F_{B0} \mid y = 0]$. Note that conditioning on event F_{B0}, Alice changes her input from $x = 0$ to $x = 1$. Intuitively, we should expect that the protocol will give output $0 \oplus 1 = 1$, not output 0. Formally:

$$\begin{aligned} p'[\textsc{out} = 0 \wedge F_{B0} \mid y = 0] &= \sum_{v \in F_{B0}} \Pr[\textsc{out} = 0 | v, x = 1, y = 0] \Pr[v | x = 0, y = 0] \\ &\leq \sum_{v \in F_{B0}} \Pr[\textsc{out} = 0 | v, x = 1, y = 0] \frac{1+\mu}{1-\mu} \Pr[v | x = 1, y = 0] \\ &\leq \frac{1+\mu}{1-\mu} \Pr[\textsc{out} = 0 | x = 1, y = 0] \leq \frac{1+\mu}{1-\mu}\, \varepsilon = \operatorname{negl}(k) \end{aligned}$$

Note that Pr in these expressions denotes the probability over an entirely honest execution of the protocol.

Next, we consider $p'[\textsc{out} = 0 \wedge \overline{F_{B0}} \mid y = 0]$. Conditioning on event $\overline{F_{B0}}$, we have that malicious Alice will in fact run the protocol honestly on input $x = 0$ during the entire interaction. So by the properties of F_{B0}, we have:

$$p'[\textsc{out} = 0 \wedge \overline{F_{B0}} \mid y = 0] \leq \Pr[\overline{F_{B0}} \mid x = 0, y = 0] \leq 3/4 + \operatorname{negl}(k)$$

Again, Pr only describes probabilities involving completely honest execution of the protocol.

Next, we consider $p'[\textsc{out} = 0 \wedge F_{B0} \mid y = 1]$. This quantity includes the event that Bob has input $y = 1$ and yet the transcript intersected the frontier at F_{B0}. Intuitively, this event should not happen very often (and less and less, as μ is larger). By the properties of F_{B0}, we have that $\beta(v, 1)/\beta(v, 0) \leq \frac{1-\mu}{1+\mu}$ for every $v \in F_{B0}$. Thus:

$$\begin{aligned} p'[\textsc{out} = 0 \wedge F_{B0} \mid y = 1] \leq \Pr[F_{B0} \mid x = 0, y = 1] &= \sum_{v \in F_{B0}} \Pr[v | x = 0, y = 1] \\ &\leq \frac{1-\mu}{1+\mu} \sum_{v \in F_{B0}} \Pr[v | x = 0, y = 0] \leq \frac{1-\mu}{1+\mu}. \end{aligned}$$

Finally, we consider $p'[\textsc{out} = 0 \wedge \overline{F_{B0}} \mid y = 1]$. Conditioning on event $\overline{F_{B0}}$, we have that malicious Alice will in fact run the protocol honestly on input $x = 0$ during the entire interaction. So by the correctness of the protocol, we have:

$$p'[\textsc{out} = 0 \wedge \overline{F_{B0}} \mid y = 1] \leq \Pr[\textsc{out} = 0 \mid x = 0, y = 1] \leq \varepsilon = \text{negl}(k).$$

Combining all of these inequalities, we finally have:

$$p'[\textsc{out} = 0] \leq \frac{1}{2}\left[\frac{3}{4} + \frac{1-\mu}{1+\mu} + \text{negl}(k)\right].$$

When μ is a fixed constant greater than $3/5$, we have that $p'[\textsc{out} = 0]$ is bounded away from $1/2$ by at least a constant, as desired. □

Uselessness of $\mathcal{F}_{\text{COIN}}$ for any SFE $\mathcal{F}$. First we consider the case when $\mathcal{F}$ is a *symmetric* SFE functionality. We use the characterization of SSFE functionalities with standalone-secure protocols from [MPR09] to show that if an SSFE functionality $\mathcal{F}$ has no standalone-secure protocol in the plain model, then either there is a standalone-secure protocol for $\mathcal{F}_{\text{EXCH}}$ in the $\mathcal{F}$-hybrid model, or else there is a frontier-based attack that violates standalone security of every purported protocol for $\mathcal{F}$ in the plain model.

In the first case, Lemma 5 demonstrates that $\mathcal{F}$ can have no standalone-secure protocol in the $\mathcal{F}_{\text{COIN}}$-hybrid world. In the second case, we observe that the frontier-based attacks go through unaltered even if the protocols are allowed access to $\mathcal{F}_{\text{COIN}}$. This is because the frontier attack merely relies on the fact that in a protocol, given a transcript prefix v, the next message depends only on one of Alice and Bob's inputs. However, this is true even if the protocol has access to $\mathcal{F}_{\text{COIN}}$— the bits from $\mathcal{F}_{\text{COIN}}$ being independent of both parties' inputs.

This allows us to conclude that in either case, there can be no protocol for $\mathcal{F}$ in the $\mathcal{F}_{\text{COIN}}$-hybrid model, giving us the following lemma

Lemma 6. *If $\mathcal{F}$ is a 2-party deterministic SSFE that has no standalone-secure (resp. UC-secure) protocol against unbounded adversaries in the plain model, then $\mathcal{F}$ has no standalone-secure (resp. UC-secure) protocol in the $\mathcal{F}_{\text{COIN}}$-hybrid model.*

Proof. To extend Lemma 5 to the case of all SSFE functionalities, we rely on results from [MPR09] (some of which are also obtained using frontier analysis).

Suppose $\mathcal{F}$ is an SSFE that has no standalone-secure protocol against unbounded adversaries (in the plain model). This class of SSFE functionalities has a combinatorial characterization from [MPR09]. From this characterization, there are three cases of $\mathcal{F}$ to consider:

(1) If $\mathcal{F}$ is decomposable but not uniquely decomposable, then we have that $\mathcal{F}_{\text{EXCH}}$ has a standalone-secure protocol in the $\mathcal{F}$-hybrid model. Thus the attack of Lemma 5 shows that $\mathcal{F}$ cannot have a protocol in the $\mathcal{F}_{\text{COIN}}$-hybrid model.

(2) If $\mathcal{F}$ is uniquely decomposable but yet has no standalone-secure protocol, then one of the frontier attacks of [MPR09] applies. In particular, [MPR09] shows that if the function evaluated is uniquely decomposable and has a certain other combinatorial property it has a standalone-secure protocol, but if it is uniquely decomposable but lacks this combinatorial property then any protocol allows either a passive (i.e., semi-honest) attack, or if not, an explicit active standalone attack.

(3) If $\mathcal{F}$ is not decomposable, then [MPR09] shows that there is in fact a *passive* attack on any protocol for $\mathcal{F}$. This attack is also constructed using frontier analysis of a purported protocol.

The attacks mentioned in (2) and (3) can be carried out as long as the protocol has the property that for any transcript v, $\Pr[v|x, y] = \alpha(v, x)\beta(v, y)$ for some functions α, β. Since this is the case for protocols in the $\mathcal{F}_{\text{COIN}}$-hybrid model, we can show that any purported protocol for $\mathcal{F}$ in the $\mathcal{F}_{\text{COIN}}$-hybrid model can be attacked in a way that violates standalone security.

In the case of UC security, the case (2) above changes, and will have a larger set of functionalities. But again, in this case if there is no passive attack on a protocol, there is an explicit attack against UC security (or even concurrent security with two sessions [MPR09]), which extends to protocols in the $\mathcal{F}_{\text{COIN}}$-hybrid model as well. Thus in the same way, the theorem holds with respect to UC security as well as standalone security. (In fact a stronger result shall be presented in Lemma 17, that even in the computationally bounded setting, $\mathcal{F}_{\text{COIN}}$ is useful for securely realizing deterministic SSFE functionalities in the UC setting only if there exists a semi-honest secure OT protocol.) □

Replacing $\mathcal{G}$ with an arbitrary selectable source. Our analysis goes through with minimal modification when $\mathcal{F}_{\text{COIN}}$ is replaced by an arbitrary selectable source. Recall that in a selectable source functionality $\mathcal{G}$, only one party can influence the output at a time (depending on which "direction" $\mathcal{G}$ is used in). When $\mathcal{G}$ is used such that only Alice influences the output, the influence on the transcript's probability can be collected into the term $\alpha(v, x)$. Similarly, when only Bob can influence the output of $\mathcal{G}$, the influence can be collected into the term $\beta(v, y)$. Therefore, we can still write $\Pr[v|x, y] = \alpha(v, x)\beta(v, y)$ for appropriate α and β. Each invocation of $\mathcal{G}$ is an atomic event with respect to the frontiers and to the adversary's changes in behavior in our our attacks.

Extending to general SFE functionalities. Finally, we prove Theorem 1, using Lemma 2. Note that a selectable source has a passive secure protocol (Alice samples an output and gives it to Bob). Thus if there exists any SFE functionality $\mathcal{F}$ for which some selectable source is useful in (UC- or standalone-) securely realizing, then by Lemma 2 selectable source is useful in (UC- or standalone-) securely realizing some SSFE functionality as well, contradicting Lemma 6.

3.4 Coins are useless for Randomized SFE

In this section, we characterize the set of randomized SFE functionalities that can be reduced to $\mathcal{F}_{\text{COIN}}$.

Since $\mathcal{F}_{\text{COIN}}$ itself is not securely realizable (in the UC or standalone model) against computationally unbounded adversaries, common randomness clearly allow more functionalities to be securely realized. In particular common randomness can be used to generate a shared sample from a publicly agreed-upon distribution. However, we show that this is essentially the only use of common randomness, when UC security is required . (When standalone security is considered, we give examples of randomized SSFE for which $\mathcal{F}_{\text{COIN}}$ is useful in a more non-trivial way in the full version.) More precisely,

Theorem 2. *A randomized SFE functionality $\mathcal{F}$ has a UC-secure protocol in the $\mathcal{F}_{\text{COIN}}$-hybrid model if and only if $\mathcal{F}$ is isomorphic to the SSFE functionality $\mathcal{F}^*$ with output function $\mathcal{F}^*$ such that $\mathcal{F}^*(x, y, r) = (h(x), r)$, where h is a deterministic function.*

Note that a secure protocol for $\mathcal{F}^*(x, y, r)$ above is simple: Alice sends $h(x)$ to Bob, and then they obtain uniformly random coins r from $\mathcal{F}_{\text{COIN}}$. Thus, any UC secure protocol for f which uses $\mathcal{F}_{\text{COIN}}$ can be replaced by one of the following form: (1) one party sends a function of its input to the other party; (2) both parties access $\mathcal{F}_{\text{COIN}}$ to obtain coins r; (3) both parties carry out local computation to produce their outputs.

Given Lemma 2, it is enough to establish our characterization for the special case of *symmetric* SFE functionalities (for which we shall denote the common output by $f(x, y, r)$).

The first step in proving Theorem 2 for SSFE is to show that only one party's input can have influence on the outcome of the other party.

Lemma 7. *If $\mathcal{F}$ is a 2-party randomized SSFE functionality with a UC-secure protocol in the $\mathcal{F}_{\text{COIN}}$-hybrid model, then $\mathcal{F}(x, y)$ is distributed as $\mathcal{F}'(x)$ (or $\mathcal{F}'(y)$), where $\mathcal{F}'$ is some randomized function of one input.*

If $\mathcal{F}$ does not have the form $\mathcal{F}'(x)$ or $\mathcal{F}'(y)$, we call it an SSFE functionality with *bidirectional influence.* We shall show later in Lemma 17 that if a bidirectional influence SSFE $\mathcal{F}$ has a UC-secure protocol in the $\mathcal{F}_{\text{COIN}}$ hybrid then there exists a semi-honest protocol for OT. However, this is not possible against computationally unbounded adversaries and hence, $\mathcal{F}$ can not have bidirectional influence.

Frontiers of influence. Henceforth, we can assume that the function $\mathcal{F}$ does not have bi-directional influence and, thus, without loss of generality, is of the form $f(x)$. Suppose we are given a protocol π for f in the $\mathcal{F}_{\text{COIN}}$-hybrid model, with simulation error ε. Without loss of generality, we assume that the last step of π is to toss a random coin which is included in the output.[5] First, define O_v^x to be the output distribution of the protocol when executed honestly on (Alice) input x, *starting from partial transcript* v. We use this to define our first frontier:

$$G = \left\{ v \;\middle|\; \begin{array}{c} \forall x', x'' : \mathrm{SD}\Big(O_v^{x'}, O_v^{x''}\Big) < \sqrt{\varepsilon} \\ \text{and no ancestor of } v \text{ satisfies the same condition} \end{array} \right\}$$

Intuitively, G represents the point at which Alice's input has first exhausted any "significant" influence on the final output distribution — her input can no longer change the output distribution by more than $\sqrt{\varepsilon}$. Next, note that the only way to induce an output distribution in the ideal world is to choose an input x according to some distribution $\mathcal{D}$ and then send x to f, yielding the output distribution $\{f(x)\}_{x \leftarrow \mathcal{D}}$. Let $\mathcal{S}$ be the space of all possible output distributions that can be induced in this way.[6] We use this to define a collection of frontiers, one for each value of x.

$$F_x = \{v \mid \mathrm{SD}(O_v^x, \mathcal{S}) > \sqrt{\varepsilon} \text{ and no ancestor of } v \text{ satisfies the same condition}\}$$

Intuitively F_x represents the first time that randomness has had a "significantly" non-trivial influence on the output when Alice's input is x. Here, the influence of randomness in the protocol is considered non-trivial if the protocol has reached a point such that the conditional output distribution induced by the protocol starting from that point cannot be achieved by Alice in the ideal world.

We now show that in a secure protocol, Alice's input must completely exhaust its influence before the randomness from $\mathcal{F}_{\text{COIN}}$ can begin to influence the output distribution.

Lemma 8. *In the above setting, let $F_x < G$ denote the event that the protocol generates a transcript that encounters frontier F_x strictly before encountering frontier G. Then $\Pr[F_x < G|x]$ is negligible for all x.*

Proof. We first observe that for complete transcripts (leaves) v, we have that $O_v^{x'} \equiv O_v^{x''}$ for all x', x''; thus G is indeed a frontier. Also, because of our normal form (last step of π is a trusted coin toss that is included in the output), every complete transcript (leaf) v satisfies $\mathrm{SD}(O_v^x, \mathcal{S}) = \Theta(1) > \sqrt{\varepsilon}$, and so F_x is indeed a

[5] To see that this is without loss of generality, define a randomized SSFE f' which on input x, outputs $f(x)$ as well as a random bit. Then define π' to be the protocol which runs π and in the last step uses $\mathcal{F}_{\text{COIN}}$ to toss a coin which is included in the output. It is easy to see that if π is a secure protocol for f, then π' is a secure protocol for f', so proving the insecurity of π' establishes the insecurity of π.

[6] Note that $\mathcal{S}$ is the space of convex combinations of $\{f(x) \mid x\}$, where here $f(x)$ denotes the discrete probability distribution itself, represented by a stochastic vector.

frontier.

Consider a particular adversary A which does the following when interacting with any environment of the appropriate form:

- Runs the protocol π honestly on input x until reaching frontier F_x. At that point, it gives the environment the value of the current partial transcript u, and pauses.
- After receiving x^* from the environment, A continues running the protocol honestly with input x^* (after back-sampling a random tape consistent with u and input x^*).

Let S denote the simulator for this adversary. If the simulator does not provide a sample $u \in F_x$ that is distributed statistically close to the real world adversary, then some environment of the required form can distinguish the real world from the ideal world. Thus, assume that the simulator S always generates u statistically close to the real world interaction A.

Consider the case where the simulator receives x^* from the environment before it has sent an input to the functionality f. Then consider the environment that sends $x^* = x$ in step 2. In this case, the real world adversary A will induce the distribution O_u^x, which is an unsimulatable distribution by the definition of F_x. No matter how the simulator subsequently chooses its input to send to f, it will induce an output distribution for the honest party whose statistical distance from O_u^x is at least $\sqrt{\varepsilon}$. Some environment of the required form can distinguish the two interactions, so we conclude that the simulator must send its input to f before step 2, except with negligible probability $\sqrt{\varepsilon}$.

Thus without loss of generality assume that S sends an input to the ideal functionality f before receiving x^* from the environment, except with negligible probability. Then consider an environment that receives $u \in F_x$, and aborts if u has a prefix in G (i.e., if $F_x \not< G$). Otherwise, the environment chooses x^* uniformly from $\{x', x''\}$, where x' and x'' are such that $\mathrm{SD}\left(O_u^{x'}, O_u^{x''}\right) \geq \sqrt{\varepsilon}$, by the definition of G. Now condition the entire interaction on the event that such an environment doesn't abort (whose probability of happening is negligibly close to $\Pr[F_x < G|x]$ in both the real and ideal interactions). Then in the ideal world, with probability at least $1/2$, the honest party's output from f will have statistical difference at least $\sqrt{\varepsilon}$ from $O_u^{x^*}$. But in the real world, the adversary always correctly induces the output distribution $O_u^{x^*}$, so some environment of this form can distinguish the real and ideal worlds with probability $O(\Pr[F_x < G|x] \cdot \sqrt{\varepsilon})$.

We conclude that $\Pr[F_x < G|x]$ must be at most $\sqrt{\varepsilon}$, which is negligible. □

Using the previous two lemmas, we can now prove the special case of Theorem 2, restricted to SSFE functionalities:

Lemma 9. *A 2-party randomized SSFE functionality $\mathcal{F}$ has a UC-secure protocol in the $\mathcal{F}_{\text{COIN}}$-hybrid model against computationally unbounded adversaries if and only if $\mathcal{F}$ is isomorphic to the SSFE functionality $\mathcal{F}^*$ with output function f^* such that $f^*(x, y, r) = (h(x), r)$, where h is a deterministic function.*

Proof. For any fundamental input x, consider the probability distribution $f(x)$, which is a corner on the convex hull of $\mathcal{S}$. By the security of the protocol, O_r^x is within statistical distance ε of $f(x)$, where r is the root of the transcript tree (the empty transcript). We also have that O_r^x is equal to the convex combination $\sum_{v \in G} \Pr[v|x] O_v^x$.

Let G^+ be the subset of G consisting of nodes v that have no ancestor in F_x. By Lemma 8, we have that $\Pr[G^+|x]$ is overwhelming. Thus O_r^x (and therefore $f(x)$) is negligibly close to the convex combination $\sum_{v \in G^+} \Pr[v|x] O_v^x$.

By the definition of G^+, each of the distributions O_v^x in the above convex combination are negligibly close (within statistical distance ε_2) to the convex space $\mathcal{S}$. A straight-forward geometric argument shows that since $f(x)$ is a corner vertex in the convex space $\mathcal{S}$, and each of the O_v^x terms in the convex combination is negligibly close to the space $\mathcal{S}$, then there is a negligible quantity δ such that the probability of encountering $v \in G$ on input x such that $\text{SD}(O_v^x, f(x)) \leq \delta$ is overwhelming. That is, almost all of the weight that x places on frontier G is placed on nodes v that induce a distribution that is negligibly close to $f(x)$.

It then follows that for any x, x' such that $f(x) \not\equiv f(x')$, the two distributions $f(x)$ and $f(x')$ are distinct vertices on the convex hull of $\mathcal{S}$. Thus their statistical distance is a constant, and so x and x' induce distributions over G that have statistical distance negligibly close to 1.

Thus consider a simple protocol ρ of the following form: Given x, Alice determines (deterministically) a sampling circuit M_x that samples the distribution $f(x)$, and sends M_x to Bob. Both parties then obtain random coins r from $\mathcal{F}_{\text{COIN}}$, and evaluate $M_x(r)$ — a sample from $f(x)$.

We claim that π is as secure as ρ in the UC sense (that is, for every adversary attacking π, there is an adversary attacking ρ that achieves the same effect in all environments). The interesting case is when a corrupt Alice is attacking π. Then the corresponding simulator does the following. It interacts with Alice in π (playing the role of an honest Bob and honest $\mathcal{F}_{\text{COIN}}$), and pauses the interaction once the G frontier has been reached. Suppose that $v \in G$ is the π-transcript so far. At this point, from the arguments above, the simulator can identify Alice's distribution $f(x)$ with only negligible error. Then the simulator sends M_x in its ρ interaction. The simulator and honest Bob toss coins in their ρ interaction to sample $z \leftarrow M_x(r)$. By the properties of G, Alice can no longer significantly bias the outcome of protocol π — the remainder of the protocol's output depends almost entirely on the ideal coin tosses. Also, with overwhelming probability, O_v^x is negligibly close to $f(x)$, so the simulator can sample a set of simulated coin tosses (for the π interaction) which will result in z as the π protocol's output. It is straight-forward to see that the simulated interaction with Alice is indistinguishable from a real interaction.

Finally, we complete the proof of Theorem 2 by observing that f is indeed isomorphic to a function of the form $g(x, y) = (g'(x), r)$, since in ρ, both parties' outputs are a function of M_x (a deterministic function of x) and independent random coins r. □

On extending to selectable sources. Unlike our results in Section 3.3, Theorem 2 does **not** generalize to arbitrary selectable sources (instead of just $\mathcal{F}_{\text{COIN}}$). To see this, one can easily construct a selectable source f which is not of the form $f(x, y, r) = (h(x), r)$. Then trivially f has a UC-secure protocol using some selectable source (namely, itself), but f is not of the form required by Theorem 2.

Indeed, to prove Theorem 2, we made a crucial distinction between Alice's choice of input influencing the output distribution and $\mathcal{F}_{\text{COIN}}$ influencing the output distribution. This distinction is lost if $\mathcal{F}_{\text{COIN}}$ is replaced by a functionality in which Alice is allowed to influence the output.

On a common random string (CRS) vs. $\mathcal{F}_{\text{COIN}}$. A common random string (CRS) is a source of shared randomness in which all random bits are generated once and for all at the beginning of a protocol interaction, rather than as-needed, as with $\mathcal{F}_{\text{COIN}}$. Our proof of Theorem 2 states that the influence of the parties' inputs ends before the influence of the shared randomness begins. Since the influence of a CRS must happen at the start of a protocol, a CRS is useless for SSFEs except those of the form $f(x, y, r) = h(x)$ (no influence from shared randomness) or $f(x, y, r) = h(r)$ (no influence from parties' inputs), for a deterministic h.

Chapter 4

Implications and Equivalences

The notion of reduction used to compare the complexity of securely realizing a task given a particular setup is closely related to the notion of security used to define the reduction. For a particular notion of reduction, as explained in our discussing in Chapter 1, there are three kinds of reductions:

1. Information theoretically true: For these reductions, there exists secure protocols for $\mathcal{F}$ in the $\mathcal{G}$-hybrid even against adversaries with unbounded computational power. Such reductions are represented as: $\mathcal{F} \sqsubseteq^{\text{STAT}} \mathcal{G}$.

2. False: There reductions are false given any computational assumption.

3. Conditionally true: The remainder of the reductions fall in the category of conditionally true reductions, which hold given some bound on the computational power of the adversaries. There reductions are represented by: $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$.

In this chapter, we are interested in coming up with tight characterization of the computational intractability assumption associated with conditionally true reductions. The computational intractability assumption associated with such a reduction is the minimal computational intractability assumption for which such a reduction will hold. But, it is important that we use a suitably strong notion of security to define our reduction. If the notion of reduction is too weak, then it might be possible that $\mathcal{F} \sqsubseteq^{\text{STAT}} \mathcal{G}$ for all functionalities $\mathcal{F}$ and $\mathcal{G}$. On the other hand, if an extremely strong notion of security is used then it might be possible that all $\mathcal{F} \sqsubseteq \mathcal{G}$ are false [MPR10b]. So, to capture sufficient diversity of computational intractability assumptions using the notion of reduction, we need to carefully choose a definition of security.

In this chapter, we shall use UC security against static corruption and we will restrict our study to two-party symmetric function evaluations. Results relevant to this chapter also appear in [MPR09] and [MPR10b]. We will crucially rely on these results. Results in [MPR09] show existence of several reductions $\mathcal{F} \sqsubseteq \mathcal{G}$ which are not true; and we will study computational intractability assumptions corresponding to these reductions. Moreover, [MPR10b] show the following results:

Lemma 10 ([MPR10b])**.** *Either $\mathcal{F} \sqsubseteq^{\text{STAT}} \mathcal{G}$,* sh-OT *assumption implies that $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ or the reduction is false.*

The above mentioned result implies that there exists a maximal assumption for the set of all computational assumptions corresponding to every reduction $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$. They also show that this is also the tightest possible result, as $\mathcal{F}_{\text{OT}} \sqsubseteq^{\text{PPT}} \mathcal{F}_{\text{COIN}}$ will imply that sh-OT assumption is true. The next result is important to show equivalence of reductions to OWF assumption.

Lemma 11 ([MPR10b])**.** *If $\mathcal{G}$ is uniquely decomposable (see [Kus89, Bea89] or Chapter 2 for the definition) then existence of on-way functions implies $\mathcal{F}_{\text{COM}} \sqsubseteq^{\text{PPT}} \mathcal{G}$.*

Results. We show that several of the reductions are in fact equivalent to OWF assumption. For example, if $\mathcal{F}$ and $\mathcal{G}$ has unique decompositions but the depth of decomposition tree of $\mathcal{G}$ is smaller than the depth of the decomposition tree of $\mathcal{F}$, then the reduction $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ is equivalent to OWF assumption. This result crucially relies on the attack proposed in [MPR09] which assumes that the adversary has unbounded computational power. In this paper, we shall show that the same attack can be simulated with the assumption that OWF assumption is false; but the simulation incurs a small error which in non-negligible but could be made arbitrarily small. Thus, instead of assuming access to a PSPACE oracle, we simulate the attack presented in [MPR09] based on the assumption that OWF assumption is false.

Similarly, we can also leverage the fact that OWF assumption is false to show that $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$, when $\mathcal{F}$ is passive trivial but not standalone trivial and $\mathcal{G}$ is standalone trivial but not UC-trivial, is equivalent to OWF assumption. Both these results crucially rely on the fact that if OWF assumption is false then distributionally one-way functions also do not exist [ILL89, Ost91, OW93]. Subsequently, this fact can be used to solve the uniform generation problem for NP statements [JVV86, BGP00] with a small non-negligible error, which can be made arbitrarily small.

We show that if $\mathcal{G}$ is exchange-like (Chapter 2) and $\mathcal{F} \not\sqsubseteq^{\text{STAT}} \mathcal{G}$ then $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ is equivalent to the sh-OT assumption. It has been shown in [MPR10b] that if $\mathcal{F}$ is not exchange-like, then $\mathcal{F}_{\text{OT}} \sqsubseteq^{\text{STAT}} \mathcal{F}$ or $\mathcal{F}_{\text{CC}} \sqsubseteq^{\text{STAT}} \mathcal{F}$. If it is the former case, then we can easily obtain a semi-honest secure protocol for oblivious transfer [PR08]. For the latter case, we shall prove that the reduction $\mathcal{F}_{\text{CC}} \sqsubseteq^{\text{PPT}} \mathcal{G}$, where $\mathcal{G}$ is exchange-like, is equivalent to sh-OT assumption. Further, we shall also show that if $\mathcal{F}$ is itself exchange-like then $\mathcal{F} \sqsubseteq \mathcal{G}$ is either unconditionally true or equivalent to sh-OT assumption. The techniques used in these results are generalizations of techniques presented in [DG03] to show that $\mathcal{F}_{\text{COM}} \sqsubseteq^{\text{PPT}} \mathcal{F}_{\text{COIN}}$ implies sh-OT assumption is true.

This statement also extends to the case when $\mathcal{G}$ is publicly-selectable source and $\mathcal{F}$ is a two-party randomized SSFE. It suffices to consider the argument when $\mathcal{G}$ is $\mathcal{F}_{\text{COIN}}$. There are two important cases to consider. If $\mathcal{F}$ is not selectable source, then the argument is similar to the previous case. Otherwise, if $\mathcal{F}$ is an oblivious-sampling functionality, then we present a different argument to show that $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{F}_{\text{COIN}}$ implies $\mathsf{sh\text{-}OT}$ assumption. Finally, if $\mathcal{F}$ is publicly-selectable source then we already know that $\mathcal{F} \sqsubseteq^{\text{STAT}} \mathcal{F}_{\text{COIN}}$ Chapter 3.

4.1 Reductions Equivalent to the OWF Assumption

Our results in this section build on the technique in [MPR09] that was used to derive the following separation in cryptographic complexity.

Lemma 12 ([MPR09])**.** *Let $\mathcal{F}$ and $\mathcal{G}$ be SSFE functionalities. If $\mathcal{F}$ has unique decomposition depth n and $\mathcal{G}$ also has unique decomposition depth $m < n$, then $\mathcal{F} \not\sqsubseteq^{\text{STAT}} \mathcal{G}$.*

In [MPR09], Theorem 12 is proven by attacking any purported protocol π for $\mathcal{F}$ in the $\mathcal{G}$-hybrid world.

First, they show (for plain protocols, not in any hybrid world) that for every adversary $\mathcal{A}$ that attacks the canonical protocol for $\mathcal{F}$, there is a corresponding adversary $\mathcal{A}'$ that attacks π, achieving the same effect in all environments. (Indeed, any functionality whose decomposition depth is at least 2 has a simple attack against its canonical protocol that violates security in the UC sense.) Intuitively, the protocol π must reveal information in the same order as the canonical protocol. More formally, at every point during the canonical protocol (say, a partial transcript t), there is a corresponding "frontier" in π — a maximal set of partial transcripts of π. If two inputs both induce transcript t in the canonical protocol (recall that it is a deterministic protocol), then they also induce statistically indistinguishable distributions on partial transcripts at the frontier. But if the two inputs do not both induce transcript t in the canonical protocol, then at the frontier they induce distributions on partial transcripts that have statistical distance almost 1. Then the adversary $\mathcal{A}'$ runs the protocol π honestly, except for occasionally "swapping" its effective input at one of these frontiers. The properties of the frontiers assure that such a swap will only negligibly affect the outcome of the interaction.

Next, to attack a protocol π in the $\mathcal{G}$-hybrid world, they imagine a plain protocol $\widehat{\pi}$ which is π composed with the canonical protocol for $\mathcal{G}$. The plain protocol $\widehat{\pi}$ has frontiers for each step of the canonical protocol (equivalently, step of the decomposition). In our setting, there are more frontiers in $\widehat{\pi}$ than there are rounds in the canonical protocol for $\mathcal{G}$, so not all the frontiers can be contained entirely within the $\mathcal{G}$-subprotocols. Thus an adversary attacking π can behave honestly in all interactions with the ideal $\mathcal{G}$, and still encounter a frontier at which to "swap" its effective input (i.e., outside of the $\mathcal{G}$-subprotocols in $\widehat{\pi}$). Indeed, there is an attack against $\mathcal{F}$ in which an adversary need only encounter one such frontier, so the protocol π is not secure.

Leveraging one-way functions. While these frontier-based attacks from [MPR09] are formulated for computationally unbounded adversaries, we show below that they can in fact be carried out under the assumption that one-way functions *do not exist*. In other words, that if a reduction exists between particular functions, then the OWF assumption is true.

These frontier-based attacks require unbounded computation because computing the frontier involves computing global statistical properties about the protocol — namely, the probability that the protocol assigns to various partial transcripts on different inputs. The attacks are otherwise effecient, so given access to an oracle that can compute these probabilities, the attack can be easily effected. In fact, these quantities need not be computed exactly for the attacks to violate security. Thus we will describe how to compute the appropriate quantities given that OWFs do not exist.

In [IL89], it is shown that the OWF assumption is implied by the much weaker assumption that *distributionally one-way* functions exist. Thus if OWFs do not exist, then no function is distributionally one-way: for every efficient function f and polynomial p, there is an efficient algorithm that on input y samples close to uniformly (within $1/p$ statistical difference) from the set $f^{-1}(y)$. We define a function related to the given protocol, and use the ability to sample its preimage to obtain a good estimate of the desired probabilities.

Theorem 3. *If $\mathcal{F}$ has unique decomposition depth n and $\mathcal{G}$ is non-trivial with unique decomposition depth $m < n$, then $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ is equivalent to the* OWF *assumption.*

Proof. First, if $\mathcal{G}$ is uniquely decomposable, then $\mathcal{F}_{\text{COM}} \sqsubseteq \mathcal{G}$ under the OWF assumption, by the argument in [MPR10b]. Then, $\mathcal{F} \sqsubseteq^{\text{STAT}} \mathcal{F}_{\text{COM}}$ since $\mathcal{F}$ is passive-trivial [MPR09]. The non-trivial direction is to show that $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ implies OWF assumption.

As described above, the attack against a protocol π for $\mathcal{F}$ in the $\mathcal{G}$-hybrid world is based on frontiers in the protocol. For a partial transcript u and inputs x for Alice and y for Bob, the probability that the protocol generates u as the prefix of its transcript can be expressed as $\alpha(u,x)\beta(u,y)$, where each of the two terms depends on only one party's input.

The frontiers used in the attack are then all defined in terms of the following quantity:

$$\eta(u, x_0, x_1) = \frac{|\alpha(u, x_0) - \alpha(u, x_1)|}{\alpha(u, x_0) + \alpha(u, x_1)}$$

or the symmetric quantity with respect to the roles of Alice & Bob. Intuitively, $\eta(u, x_0, x_1)$ measures how correlated the transcript u is to Alice's input being x_0 versus x_1. In fact, the entire frontier-based attack can be carried out in polynomial time given an oracle that answers questions of the form "Is $\eta(u, x_0, x_1) \geq 1 - \nu(k)$?", where ν is a certain negligible function in the security parameter. If instead the oracle can answer questions of this form where $\nu(k) = 1/k^c$ for a chosen constant c, then the adversary's attack may fail with at most an extra 1/poly factor. All the attacks from [MPR09] demonstrate that the real and ideal worlds can be distinguished with constant bias, so they can indeed tolerate this additional 1/poly slack factor. Thus it suffices to show how to implement such an oracle.

We compute $\eta(u, x_0, x_1)$ as follows: First, Consider the function $f(x, r_A, y, r_B, i) = (\tau, x)$, where τ is the first i bits of the transcript produced by the protocol when executed honestly on inputs (x, y), where r_A and r_B are the random tapes of Alice and Bob, respectively. We use the guarantee of no distributionally one-way functions to sample from $f^{-1}(u, x_0)$ and $f^{-1}(u, x_1)$. If both preimages are empty, then the protocol never generates u as a partial transcript on inputs x_0 or x_1. If only one is empty, then $\eta(u, x_0, x_1) = 1$.

Otherwise, assume u is indeed a possible partial transcript for both x_0 and x_1 (i.e., the protocol assigns positive probability to u when Alice has inputs x_0 or x_1). Our previous sampling of f^{-1} has yielded an input y^* such that u is a possible partial transcript when executing π on inputs (x_0, y^*). Thus u is also a possible partial transcript on inputs (x_1, y^*). Now define:

$$g(x, r_A, y, r_B, i) = \begin{cases} (\tau, y) & \text{if } x \in \{x_0, x_1\} \\ \bot & \text{otherwise} \end{cases}$$

We now sample n times from $g^{-1}(u, y^*)$. Let n_i be the number of times the sampled preimage included x_i as the first component. Then $|n_0 - n_1|/n$ is an estimate of $\eta(u, x_0, x_1)$. By setting n to be a sufficiently large polynomial in the security parameter, we can ensure that the estimate is within an additive factor $1/k^c$ of the actual value, with high probability. □

Theorem 4. *If $\mathcal{F}$ is passive-trivial but not standalone-trivial and $\mathcal{G}$ is standalone-trivial but not UC-trivial, then $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ is equivalent to the* OWF *assumption.*

Proof. The fact that $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ under the OWF assumption is by the same argument as in the previous proof, because standalone trivial two-party deterministic SSFE have unique decomposition.

For the other direction, suppose π is a secure protocol for $\mathcal{F}$ in the $\mathcal{G}$-hybrid world. Standalone secure protocols *for SFE* functionalities are closed under composition. Thus we have a standalone-secure protocol π' for $\mathcal{F}$ without any trusted party.

Being passive-trivial, $\mathcal{F}$ is surely decomposable, and we consider two cases. When $\mathcal{F}$ is uniquely decomposable, then [MPR09] showed that in the *unbounded* setting, for every adversary $\mathcal{A}$ attacking the canonical protocol, there is an adversary $\mathcal{A}'$ attacking π' such that no environment can distinguish between the two interactions. When $\mathcal{F}$ is uniquely decomposable but not standalone-trivial, there is a simple attack against the canonical protocol for $\mathcal{F}$ that violates standalone security with constant probability. Thus translating this attack into an efficient (assuming that OWF assumption is false) attack on π' using the techniques described in the previous proof, we see that π' is not standalone-secure; a contradiction.

On the other hand, if $\mathcal{F}$ is not uniquely decomposable, then $\mathcal{F}_{\text{EXCH}} \sqsubseteq^{\text{STAT}} \mathcal{F}$ via a simple protocol. As such, by composing several protocols, we obtain a standalone-secure protocol π for $\mathcal{F}_{\text{EXCH}}$. Consider an interaction using π in which the honest party choses an input at random. We describe an attack that can be carried out assuming that the OWF assumption is false, which biases the honest party's output towards 0 by a noticeable amount (this is the same standalone attack presented in Chapter 3):

At each partial transcript u, consider $\eta(u, 0, 1)$ (which measures the transcript's bias towards Alice's input 0 or 1, defined in the previous proof) At the beginning of the protocol, the value of this function is 0, and at the end of the protocol, it is negligibly close to 1 with overwhelming probability since the protocol results in Bob learning Alice's input.

Similarly, define $\eta'(u, 0, 1)$ as a transcript's bias towards Bob's input. By symmetry, with probability at least 1/2, the partial transcript achieves $\eta(u, 0, 1) > 1/2$ before it achieves $\eta'(u, 0, 1) > 1/2$. Thus an attack for Bob is to discover via the sampling procedure described above the first point at which $\eta(u, 0, 1) > 1/2$ but $\eta'(u, 0, 1) \leq 1/2$. At that point, Bob switches his input to match Alice's, in order to bias the output towards 0. Bob reaches such a point with probability at least 1/2, Since $\eta'(u, 0, 1) \leq 1/2$, the correctness of the protocol implies that Bob's output will be 0 with overwhelming probability. Thus this attack successfully biases the output towards 0 with bias 1/4 minus some inverse polynomial in the security parameter. □

4.2 Reductions Equivalent to the $\mathsf{sh\text{-}OT}$ Assumption

Recall that a two-party SSFE $\mathcal{F}$ is exchange-like if $\mathcal{F} = \mathcal{F}^{i,j}_{\text{EXCH}}$ for some i, j.

Lemma 13 ([MPR10b]). *If $\mathcal{F}$ is not exchange-like, then either $\mathcal{F}_{\text{OT}} \sqsubseteq^{\text{STAT}} \mathcal{F}$ or $\mathcal{F}_{\text{CC}} \sqsubseteq^{\text{STAT}} \mathcal{F}$.*

The proof is a simple combinatorial characterization. If $\mathcal{F}$ is not exchange-like, then it contains one of two kinds of 2×2 minors. One of these minors yields an unconditional $\mathcal{F}_{\text{OT}}$ protocol, due to a result of [KM11]. The other kind of minor yields an elementary protocol for $\mathcal{F}_{\text{CC}}$.

Our main classification involving exchange-like functionalities is the following:

Theorem 5. *If $\mathcal{G}$ is exchange-like and non-trivial, then either $\mathcal{F} \sqsubseteq^{\text{STAT}} \mathcal{G}$, or $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ is equivalent to the* sh-OT *assumption.*

Proof. From [MPR10b], we have that $\mathcal{G}$ is $\sqsubseteq^{\text{PPT}}$-complete under the sh-OT assumption, since it is non-trivial. Thus $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ under the sh-OT assumption.

For the other direction, we break the proof into two parts, depending on the status of $\mathcal{F}$. These are carried out in the following two lemmas. □

Lemma 14. *If $\mathcal{F}$ is not exchange-like, and $\mathcal{G}$ is exchange-like and non-trivial, then $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ implies the* sh-OT *assumption.*

Proof. Given that $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$, we directly construct a passive secure protocol for $\mathcal{F}_{\text{OT}}$. From [MPR10b], we have that $\mathcal{F}_{\text{OT}} \sqsubseteq^{\text{STAT}} \mathcal{F}$ or $\mathcal{F}_{\text{CC}} \sqsubseteq^{\text{STAT}} \mathcal{F}$. Thus, $\mathcal{F}_{\text{OT}} \sqsubseteq^{\text{PPT}} \mathcal{G}$ or $\mathcal{F}_{\text{CC}} \sqsubseteq^{\text{PPT}} \mathcal{G}$ by the universal composition theorem.

In the first case, $\mathcal{F}_{\text{OT}}$ has the property that any UC-secure protocol for $\mathcal{F}_{\text{OT}}$ (even in a hybrid world) is also itself a semi-honest-secure protocol [PR08]. $\mathcal{G}$ also has a semi-honest-secure protocol (namely, its canonical protocol since it is decomposable). Composing these two protocols yields a semi-honest (plain) protocol for $\mathcal{F}_{\text{OT}}$.

In the other case, suppose π is the secure protocol for $\mathcal{F}_{\text{CC}}$ in the $\mathcal{G}$-hybrid world. Recall that $\mathcal{F}_{\text{CC}}$ has a function table $\boxed{\begin{smallmatrix}0 & 2\\ 1 & 2\end{smallmatrix}}$, which we interpret as Alice sending a bit (top row or bottom row), Bob choosing whether or not to recieve it (left column or right column), and Alice learning Bob's choice (whether or not the output was 2). We directly use π to construct a semi-honest $\mathcal{F}_{\text{OT}}$ protocol as follows, with Alice acting as the OT sender (with inputs x_0, x_1) and Bob the receiver (with input b):

- The parties instantiate two parallel instances of π, with Alice acting as the sender. Since there is no access to an external $\mathcal{G}$, Bob will simulate Alice's interface with instances of $\mathcal{G}$— that is, Alice will send her $\mathcal{G}$ inputs directly to Bob, and he will give simulated responses from instances of $\mathcal{G}$. Alice sends bit x_0 in the first instance, and x_1 in the second instance, running the protocol honestly.

- In protocol instance $(1-b)$, Bob carries out the simulation of $\mathcal{G}$-instances and the π protocol completely honestly. He runs the π protocol on the input that does not reveal Alice's input.

- In protocol instance b, Bob honestly runs the UC simulator for π, treating Alice as the adversary (including simulating Alice's interface with $\mathcal{G}$-instances). At some point, the simulator extracts Alice's bit x_b to send to $\mathcal{F}_{\text{CC}}$. Bob continues running the simulator as if $\mathcal{F}_{\text{CC}}$ responded with output 2. When the interaction completes, Bob outputs x_b.

By the UC security of π, Alice's view is computationally independent of b (i.e., she cannot distinguish an interaction with π's simulator from an interaction in which the receiver and $\mathcal{G}$ are honest). Bob correctly learns x_b, and we must argue that he has no advantage guessing x_{1-b}. If all $\mathcal{G}$-instances were external to the $(1-b)$ interaction as ideal functionalities, then the security of π would imply that Bob has no advantage in guessing x_{1-b} after running the protocol with the input that does not reveal Alice's bit. Being an exchange function, $\mathcal{G}$ has the property that Bob always learns all of Alice's inputs. Thus Alice can send her $\mathcal{G}$-inputs directly to Bob, without any affect on the security of the protocol. This is exactly what happens in the $(1-b)$ interaction. □

For the case where $\mathcal{F}$ is exchange-like, we completely characterize when $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ is equivalent to the sh-OT assumption.

Lemma 15. *Let $\mathcal{F}$ and $\mathcal{G}$ be exchange-like, so without loss of generality, $\mathcal{F} = \mathcal{F}_{\text{EXCH}}^{i,j}$ and $\mathcal{G} = \mathcal{F}_{\text{EXCH}}^{i',j'}$. Then if $i \leq i'$ and $j \leq j'$, or if $i \leq j'$ and $j \leq i'$, then $\mathcal{F} \sqsubseteq^{\text{STAT}} \mathcal{G}$. Otherwise, $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ is implies the* sh-OT *assumption.*

Proof. The protocol to show $\mathcal{F} \sqsubseteq^{\text{STAT}} \mathcal{G}$ is elementary. To perform an i, j exchange using $\mathcal{G}$, simply place $\mathcal{G}$ in the appropriate send inputs directly to $\mathcal{G}$ (with Alice and Bob exchanged if necessary). Each party aborts if the other party provided an input to the (i', j') exchange which was out of bounds for an (i, j) exchange. The security of this protocol is straight-forward.

We sketch here the main ideas behind proving the other direction. The full proof is given in the appendix. For simplicity, suppose that $\mathcal{F} = \mathcal{F}_{\text{EXCH}}^{i,i}$ and $\mathcal{G} = \mathcal{F}_{\text{EXCH}}^{(i-1),(i-1)}$.

Suppose we have a protocol π demonstrating $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$. The role of the simulator for π is to first extract the input of a corrupt party, send it to $\mathcal{F}$ in the ideal world, and then continues to simulate π consistently given the output from $\mathcal{F}$.

Again for simplicity, suppose that the simulator for a *passively* corrupt Alice always extracts during round r_A.[1] Then through $r_A - 1$ rounds of the simulation, Alice's view is independent of Bob's input. If Bob's input is random (uniform in $[i]$), then after round r_A, Alice cannot guess Bob's input with probability greater than $\zeta = (i-1)/i$, since there are only $i-1$ possible responses from the simulated $\mathcal{G}$ that the simulator can give to complete the round. By the soundness of the simulation, an honest Alice cannot predict Bob's input with probability greater than $\zeta + \text{negl}(k)$ after r_A rounds of an *honest* interaction with Bob. Similarly, if the simulator for a passively corrupt Bob always extracts during round r_B, then an honest Bob cannot predict

[1] If a round *begins* with a call to the external functionality $\mathcal{G}$, then the round concludes when the parties receive their output from this external functionality. Extracting *during* round r means that the simulator extracts after seeing the adversary's input to the external functionality, and before delivering the corresponding output.

Alice's random input with probability greater than $\zeta + \text{negl}(k)$ after r_B rounds of an honest interaction with Alice.

By symmetry, suppose that $r_A \leq r_B$. Then a semi-honest protocol for a weak variant of OT is as follows:

- Alice chooses two random elements $x_0, x_1 \in [i]$ and runs two instances of the protocol π with these respective inputs, for r_A rounds.

- Bob's input is a choice bit $b \in \{0,1\}$, and in the bth interaction with π, Bob runs the simulator for π against Alice (including simulating her interface with instances of $\mathcal{G}$). In the $(1-b)$ interaction, Bob runs the π protocol honestly on a fixed input, and also honestly simulates all instances of $\mathcal{G}$. After r_A rounds, the b-interaction successfully extracts x_b, which Bob outputs.

By the security of the protocol π, Alice cannot distinguish between the b and $(1-b)$ instances. In the $(1-b)$ instance, Bob runs the protocol honestly against Alice for $r_A \leq r_B$ rounds, and as such, cannot predict x_{1-b} with probability greater than $\zeta + \text{negl}(k)$. Using a standard amplification technique (Appendix A.2), we can obtain a full-fledged OT protocol in which Bob has no advantage in predicting x_{1-b}.

The main proof is more involved in several ways. First, the case where the dimensions of $\mathcal{F}$ and $\mathcal{G}$ are incomparable requires a more careful analysis. Second, r_A and r_B need not be fixed rounds, but may be random variables. In this case, the parties must essentially guess $\min\{r_A, r_B\}$. Still, we can obtain a weak OT protocol in which Bob has noticeable uncertainty about x_{1-b}, and which is therefore amenable to amplification. □

Using an analogous approach, we also show the following in the appendix:

Lemma 16. *$\mathcal{F}^{2,2}_{\text{EXCH}} \sqsubseteq \mathcal{F}_{\text{COIN}}$ is equivalent to the* sh-OT *assumption.*

4.2.1 Reductions to Publicly-selectable Source

In this section we shall prove the following theorem:

Theorem 6. *Let $\mathcal{F}$ be a two-party randomized SSFE and $\mathcal{G}$ be a publicly-selectable source. Either $\mathcal{F} \sqsubseteq^{\text{STAT}} \mathcal{G}$ or $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ is equivalent to* sh-OT *assumption.*

The result reduces to two main cases: $\mathcal{F}$ has bidirectional influence or not; each of which are dealt in the following sections.

4.2.2 Hardness of SSFE Functionalities with Bidirectional Influence

In this section we will show the following result:

Lemma 17. *Let $\mathcal{F}$ be a (possibly randomized) SSFE, with bidirectional influence. For any publicly-selectable source $\mathcal{G}$, $\mathcal{F}$ has a UC secure protocol in the $\mathcal{G}$-hybrid then there exists a semi-honest secure protocol for OT.*

This generalizes a result we considered earlier, where we showed the same result when $\mathcal{F}$ was a two-party deterministic SSFE. We will sketch the outline of the ideas of the major modifications and the interested reader is requested to refer to the original proof in the appendix.

Consider the following protocol $P_{A\rightarrow B}$: Suppose Alice has two inputs x_0, x_1 from her input domain of $\mathcal{F}$ and Bob has a choice bit b. There are two session S_0 and S_1. In session S_b, Bob runs the simulator for corrupt Alice and in session S_{1-b} Bob runs the protocol honestly. Bob aborts both sessions at a round which is uniformly chosen at random. The instance of $\mathcal{G}_A$ and $\mathcal{G}_B$ are realized by Alice and Bob, respectively, computing the output honestly and sending it to the other party. If in session S_b, Bob is unable to extract Alice's input x_b, then it asks Alice to send both her inputs; and Alice sends (x_0, x_1) to Bob.

There is a similar protocol where the roles of Alice and Bob are reversed, say $P_{B\rightarrow A}$. It has been shown in [MPR10a] that under certain guarantees, the protocols $P_{A\rightarrow B}$ and $P_{B\rightarrow A}$ can be amplified into semi-honest secure protocols for OT. We will show that if $\mathcal{F}$ has a UC secure protocol in the $\mathcal{G}$-hybrid then the conditions are satisfied for $P_{A\rightarrow B}$ or $P_{B\rightarrow A}$.

Since $\mathcal{F}$ is bidirectional, for every Alice input x and x' there exists a Bob input y such that the distributions $f(x, y)$ and $f(x', y)$ are different. Similarly, for every Bob input y and y' there exists an Alice input x such that the distributions $f(x, y)$ and $f(x, y')$ are different. Consider the case when Alice is semi-honest corrupt. Let t_A be the round where Alice can predict Bob's input with probability at least $\zeta = 1/n + c < 1$, where the size of Bob's input domain is n and c is a small constant. Suppose in round s_B, the simulator for corrupt Alice extracts her inputs and sends it to $\mathcal{F}$. The simulator extracts the correct input of Alice, otherwise there exists a Bob input which can distinguish the actual input of Alice from the input sent by the simulator. Since Bob input is chosen uniformly at random, the environment can distinguish these two case with constant probability. We claim that $t_A \geq s_B + 1$. Suppose s_B is a round where Alice sends a bit or they use $\mathcal{G}_A$. In this case, at this round, all inputs for Bob are equally likely and hence $t_A > s_B$. Otherwise, if s_B is a round where Bob sends a bit or they use $\mathcal{G}_B$, then the simulator could have alternatively extracted one round earlier. This reduces the problem to the previous case.

In particular, we can conclude that $E[t_A] \geq E[s_B] + 1$. Let u_A be the round where Alice in the real protocol can predict Bob's input with probability ζ. Security guarantee implies that the simulated view should not be significantly different from the real view. Hence we obtain that $|E[u_A] - E[t_A]| \leq \varepsilon/\zeta = \varepsilon'$. This implies that $E[u_A] \geq E[s_B] + (1 - \varepsilon')$.

Similarly, we define the quantities s_A, t_B and u_B and conclude that $E[u_B] \geq E[s_A] + (1 - \varepsilon')$. These two inequalities imply that either $E[u_A] \geq E[s_A] + (1-\varepsilon')$ or $E[u_B] \geq E[s_B] + (1-\varepsilon')$. In other words, either the simulator for corrupt Bob extracts significantly before Alice has a good guess about Bob's input; or other way around. Using the algorithm mentioned earlier, this guarantee is sufficient to obtain a semi-honest secure protocol for oblivious-transfer.

4.2.3 Case of Oblivious Sampling

In this section we prove the following result:

Theorem 7. *Let $\mathcal{F}$ be a selectable source which is not publicly-selectable source, then $\mathcal{F}$ reduces to $\mathcal{F}_{\text{COIN}}$ is equivalent to* sh-OT*.*

But first we prove the following simple lemma that will be useful in proving the above mentioned theorem.

Lemma 18. *For any publicly-selectable source $\mathcal{G}$, in the computationally unbounded (as well as PPT) setting, $\mathcal{G}$ reduces to $\mathcal{F}_{\text{COIN}}$; also, $\mathcal{F}_{\text{COIN}}$ reduces to $\mathcal{G}$, unless $\mathcal{G}$ is trivial.*

Proof. W.l.o.g, let Alice be the party whose input may have influence on the output in $\mathcal{G}$. Let $\mathcal{D}$ denote the set of output distributions for non-redundant inputs for Alice. Note that since $\mathcal{G}$ is a publicly-selectable source, the distributions in $\mathcal{D}$ have disjoint supports.

$\mathcal{G}$ reduces to $\mathcal{F}_{\text{COIN}}$: this follows from a simple protocol for $\mathcal{G}$ as follows (we omit the routine security analysis):

- On input x, Alice determines the unique convex combination of distributions in $\mathcal{D}$ that equals the output distribution for x. The uniqueness is a consequence of those distributions having disjoint supports.
- Alice samples an element from $\mathcal{D}$ according to its weight in the above convex combination, and announces it. (We remark that a cheating Alice could use any strategy to choose an element from $\mathcal{D}$; however it can be mapped to simply choosing an input and then following the protocol honestly.)
- Alice and Bob obtain coins from $\mathcal{F}_{\text{COIN}}$, and use them to sample an outcome from the announced distribution.

$\mathcal{F}_{\text{COIN}}$ reduces to $\mathcal{G}$, unless $\mathcal{G}$ is trivial: $\mathcal{G}$ is trivial iff every distribution in $\mathcal{D}$ has zero entropy. Otherwise the following is a secure protocol[2] for $\mathcal{F}_{\text{COIN}}$ using $\mathcal{G}$ (again, we omit the standard security analysis). Briefly,

[2] Note that in a secure realization for $\mathcal{F}_{\text{COIN}}$ (without guaranteed output delivery) either party is allowed to abort the protocol, possibly after seeing the outcome of the protocol. This is the standard UC security guarantee for 2-party functionalities (and more generally, when there is no honest majority assumption).

in this protocol the parties apply a von Neumann extractor to the outcome sampled from $\mathcal{G}$, to obtain a fair coin.

- Let x be a fixed non-redundant input for Alice such that the output distribution for x is in $\mathcal{D}$ and has positive entropy. Let $Z_0 \subseteq Z$ be a subset of the outcomes so that for input x, the probability that the outcome is in Z_0 is p, $0 < p < 1$. Let $Z_1 = Z \backslash Z_0$.
- Alice and Bob repeat the following until they are "satisfied":
  - Alice sends x to $\mathcal{G}$ twice.
  - If in either instance, the output from $\mathcal{G}$ is not from the support of the distribution corresponding to x, Bob aborts the protocol. Note that since $\mathcal{G}$ is a publicly-selectable source, this essentially forces Alice to either send an input equivalent to x or probabilistically abort.
  - Else, if exactly one of the outputs is from Z_0 and one from Z_1 then the parties are satisfied
- If the outputs in the last pair of invocations of $\mathcal{G}$ where in Z_0 and Z_1 respectively, the common output is 0; else (the outputs where in Z_1 and Z_0 respectively) the common output is 1.

Since p is a constant independent of the security parameter, this protocol runs in expected constant number of rounds, and except with negligible probability, ends in a polynomial number of rounds. □

Proof of Theorem 7

We start with the following classification of 2-party (randomized) SSFE functionalities.

Lemma 19. *Every 2-party SSFE functionality falls into one of the following categories.*

1. *UC-reduces to $\mathcal{F}_{\text{COIN}}$ (in the computationally unbounded setting).*
2. *An oblivious sampling functionality.*
3. *A functionality with bi-directional influence.*

Proof Sketch. This follows from the partitioning of SSFE functionalities into (a) uninfluenced functionalities, (b) functionalities with unidirectional influence, and (c) those with bidirectional influence (see Chapter 2). If $\mathcal{F}$ is publicly-selectable source, then it statistically reduces to $\mathcal{F}_{\text{COIN}}$. If $\mathcal{F}$ is selectable source which is not publicly-selectable source then it is oblivious sampling. Finally, if $\mathcal{F}$ is not a selectable source, then it has bidirectional influence. □

Given the above classification we prove Theorem 5 by considering functionalities in each of the above categories separately.

- *Category 1.* Since $\mathcal{F}$ in this category is UC-reducible to $\mathcal{F}_{\text{COIN}}$ in the computationally unbounded setting, the condition in the theorem is satisfied.
- *Category 2.* If $\mathcal{F}_{\text{COIN}}$ is useful for UC-securely realizing a functionality $\mathcal{F}$ in this category, and therefore in particular $\mathcal{F}$ UC-securely reduces to $\mathcal{F}_{\text{COIN}}$, then below we shall give a semi-honest secure OT protocol.
- *Category 3.* Previously, it has already been shown that if a functionality in this category reduces to $\mathcal{F}_{\text{COIN}}$, then there exists a semi-honest secure OT protocol.

Thus to complete the proof of Theorem 5 it remains to show the following.

Lemma 20. *If an oblivious sampling functionality $\mathcal{F}$ has a UC-secure protocol in the $\mathcal{F}_{\text{COIN}}$-hybrid model, then there exists a semi-honest secure OT protocol.*

Proof. Since $\mathcal{F}$ is an oblivious sampling functionality, it is an SSFE functionality $\mathcal{F}_f$ with unidirectional influence (w.l.o.g, assume that Alice's input influences Bob's output) such that there exist two non-redundant inputs $x_0, x_1 \in X$ and an output $z \in Z$, such that the distributions $f(x_0) \neq f(x_1)$ and z falls in the intersection of the supports of $f(x_0)$ and $f(x_1)$.

Suppose Π is a protocol in $\mathcal{F}_{\text{COIN}}$-hybrid that securely realizes $\mathcal{F}$. Before we specify and analyze our protocol, we elaborate on what it means for Π to securely realize $\mathcal{F}$. Let $\mathcal{S}_{\Pi}^{A}$ be the simulator for a corrupt Alice, such that no environment can distinguish between Alice being in an execution of Π and Alice being in an execution simulated by $\mathcal{S}_{\Pi}^{A}$. (Similarly, let $\mathcal{S}_{\Pi}^{B}$ be the simulator for corrupt Bob.) Then $\mathcal{S}_{\Pi}^{A}$ behaves as follows: it interacts with corrupt Alice simulating to her Bob's messages in Π, while also interacting with the ideal functionality $\mathcal{F}$ playing Alice's role. At some point $\mathcal{S}_{\Pi}^{A}$ would send an input to $\mathcal{F}$ on behalf of Alice, and obtain an outcome (which Bob also obtains and outputs to the environment). We use the following observation about the input that $\mathcal{S}_{\Pi}^{A}$ sends to $\mathcal{F}$, when corrupt Alice follows the protocol Π honestly. Here, two inputs x and x' are called equivalent if the distributions $f(x)$ and $f(x')$ are identical.

Claim 1. *Consider the ideal execution involving a corrupt Alice, $\mathcal{S}_{\Pi}^{A}$ and the ideal functionality $\mathcal{F}$. If corrupt Alice follows Π honestly using a non-redundant input x, then the input that $\mathcal{S}_{\Pi}^{A}$ sends to $\mathcal{F}$ is, except with negligible probability, equivalent to x.*

Proof. Let $\alpha_{x'}$ be the probability with which $\mathcal{S}_{\Pi}^{A}$ sends the input x' to $\mathcal{F}$. Then the resulting output distribution is $\sum_{x' \in X} \alpha_{x'} f(x')$. However, for the simulation to be good, we require this to be negligibly different from $f(x)$. Consider the set X' of all inputs not equivalent to x. Since x is not redundant, $f(x)$ lies outside the convex hull of the set of distributions $\{f(x')|x' \in X'\}$. Since the probabilities are constant (independent of the security parameter), the Euclidean distance between $f(x)$ (considered a point in the space $\mathbb{R}^{|Z|}$) and this convex hull is some constant, say ℓ. Then, the distribution $\sum_{x' \in X} \alpha_{x'} f(x')$ has a Euclidean distance of at least $\ell(\sum_{x' \in X'} \alpha_{x'})$ from $f(x)$. Since this distance must be negligible (as the Euclidean distance is at most twice the statistical distance), and ℓ is constant, it must be that $\sum_{x' \in X'} \alpha_{x'}$ is negligible. In other words, except with negligible probability $\mathcal{S}^A_\Pi$ sends an input equivalent to x, completing the proof of the claim. □

To show that there exists a semi-honest secure protocol for OT, we shall show that there is such a protocol for the functionality $\mathcal{F}_{\text{AND}}$, which takes a bit each from Alice and Bob and outputs their logical AND to Bob (Alice gets an empty output). (This is enough since it is easy to see that in the semi-honest case OT reduces to $\mathcal{F}_{\text{AND}}$.) Consider the following protocol for $\mathcal{F}_{\text{AND}}$.

```
Let Alice's input be $x^* \in \{0,1\}$ and Bob's input be $y^* \in \{0,1\}$.

For $i = 1$ to $k$

    Until Alice and Bob are "satisfied"

        Alice picks $b_i \leftarrow \{0,1\}$, and executes $\Pi$ with Bob, with Bob implementing $\mathcal{F}_{\text{COIN}}$, with input
        $x^i := x_{b_i}$

        If $y^* = 0$, then

            Bob executes the protocol $\Pi$ with Alice, implementing $\mathcal{F}_{\text{COIN}}$ himself, and obtains output
            $\hat{z}$.

        Else ($y^* = 1$),

            Bob runs the simulator $\mathcal{S}^A_\Pi$ for a corrupt Alice in $\Pi$, until the simulator extracts an input
            $\hat{x}^i$; the simulator expects a response from $\mathcal{F}$ on sending this input to it.

            Bob samples $\hat{z}$ from $f(\hat{x}^i)$, and feeds this back to the simulator as the output from $\mathcal{F}$.

            Bob continues executing the simulator until the end of the protocol.

        If $\hat{z} = z$ then Alice and Bob are satisfied, else not.

Alice sends $w = x^* \oplus b_1 \oplus b_2 \oplus \ldots \oplus b_k$ to Bob.

If $y^* = 0$ Bob outputs 0, else he outputs $w \oplus \hat{b}_1 \oplus \hat{b}_2 \oplus \ldots \oplus \hat{b}_k$, where the bit $\hat{b}_i$ is 0 iff $\hat{x}^i = x_0$.
```

We shall argue that if Π is a secure protocol for $\mathcal{F}$, then this protocol is a semi-honest secure protocol for $\mathcal{F}_{\text{AND}}$ in the PPT setting.

Firstly, we show that the protocol is correct: for any pair of inputs, the outputs of the protocol is the same as that of the ideal functionality $\mathcal{F}_{\text{AND}}$. Alice produces an empty output in the protocol and in the ideal execution. When $y^* = 0$, Bob's output is 0 in both cases. It only remains to analyze the case when $y^* = 1$. For this case, we argue that in the protocol, $\hat{x}^i = x^i$ for all i, so that Bob's ouput is indeed x^* as it will be in the ideal execution. This follows from Claim 1.

It is enough to consider the case when exactly one of Alice and Bob is passively corrupt.

Given the correctness of the protocol, it remains to show that the view of the corrupt party can be simulated based on the corrupt party's input and output (and given those, independent of the input of the other party).

If Alice is corrupt, consider a simulator which simply runs our protocol with Bob's input set to (say) 0, and sends Alice's input to $\mathcal{F}_{\text{AND}}$. By the correctness of the protocol, we need only argue that the view of Alice is nearly the same as in the simulation for $y^* = 0$ and $y^* = 1$. Clearly this is true when $y^* = 0$. On the other hand, Alice's view is nearly identical when $y^* = 1$ and $y^* = 0$ by the indistinguishability guarantee of the simulator $\mathcal{S}_{\Pi}^{A}$.

If Bob is corrupt, consider the following (semi-honest) simulation. If $y^* = 1$, then the simulator sends 1 to $\mathcal{F}_{\text{AND}}$ and obtains x^* in response; then it faithfully runs our protocol with Alice's input set to x^*. If $y^* = 0$, then the simulator obtains no information from $\mathcal{F}_{\text{AND}}$; in this case it simply picks an arbitrary input for Alice, say 0, and runs our protocol faithfully. Note that this has the effect that the last message sent in the protocol when $x^* = 1$ could be wrongly distributed. However we argue that the last message when $x^* = 1$ is nearly identically distributed as when $x^* = 0$, conditioned on Bob's view in the rest of the protocol. For this, we first replace each execution of Π in our protocol as well as in our simulation with a simulation using $\mathcal{S}_{\Pi}^{B}$ interacting with an instance of the ideal functionality $\mathcal{F}$. This causes a negligible change in the two distributions. Then, for an execution of Π, conditioned on Bob's view (in which the only information about each b_i is the fact that the response from the ideal functionality $\mathcal{F}$ is z), $p := \Pr[b_i = 0] = \Pr[f(x_0) = z]/(\Pr[f(x_0) = z] + \Pr[f(x_1) = z])$, and $\Pr[b_i = 1] = 1 - p$ (independently for each i), for some constant (i.e., independent of k) p, with $0 < p < 1$. Then $|\Pr[\bigoplus_{i=1}^{k} b_i = 0] - \Pr[\bigoplus_{i=1}^{k} b_i = 1]| = |(p - (1-p))^k|$ is negligible, or in other words $\bigoplus_{i=1}^{k} b_i$ is close to a uniformly distributed bit. Thus the last message sent out by Alice is nearly identically distributed for $x^* = 0$ and $x^* = 1$. □

Chapter 5

Weak Coin Tossing

A fundamental problem in cryptography is to design protocols that allow two mutually distrusting parties to agree on a random coin. The problem of coin flipping is certainly intrinsically fascinating, but moreover coin flipping protocols have proven to be extremely useful to the theory and design of secure protocols: For example, they are an essential ingredient in all known secure two-party and multi-party computation protocols (e.g. Goldreich, Micali, and Wigderson [GMW87]). They have also proven to be influential more widely: For example, they provide a primary motivation for the utility of the Common Random String (CRS) model [BFM88], one of the most popular models for cryptographic protocol design.

The problem of coin flipping was introduced in the seminal work of Blum [Blu82], who described the task by means of the following scenario: Alice and Bob are divorcing, and have agreed to let the ownership of their favorite car be decided by a coin toss: Heads means that Alice gets the car, and Tails means that Bob gets it. Unfortunately, Alice and Bob are not willing to be in the same room, and need to implement this coin toss over the telephone. As such, Alice and Bob want a protocol such that (informally speaking):

1. The transcript of their conversation uniquely determines who gets the car.

2. If both Alice and Bob behave honestly, then Alice and Bob should both get the car with probability $1/2$.

3. If Alice behaves maliciously but Bob behaves honestly, then Alice cannot significantly increase the probability that she gets the car. Similarly, if Bob behaves maliciously but Alice behaves honestly, then Bob cannot significantly increase the probability that he gets the car.

Such a protocol incentivizes honest behavior by both parties, since they know that deviating from the protocol would not allow them to obtain any significant gain. Here, we are making the non-trivial assumption that both parties *want* to get the car – that is, we do not disallow a cheating Alice to increase the probability of Bob getting the car[1]. As such, this notion of coin flipping is often called "weak" coin flipping (explicitly

[1] Note that by symmetry, our requirements imply that if a protocol fails to meet the requirements, it must be the case that either (1) a single party can bias the outcome significantly in both directions, or (2) both parties can bias the outcome significantly in the *same* direction. If neither of these attacks are possible, then there is always a renaming of Alice and Bob that implies that the protocol meets our requirements.

in the quantum cryptography literature [KN04]), in contrast to "strong" coin flipping where neither party should be able to bias the coin significantly in either direction. [2] Of course, a crucial parameter here is how much bias constitutes a "significant gain." Let us define a $(1-\delta)$-secure weak coin flipping protocol to be one where no cheating party can increase the probability of their desired outcome by more than an additive factor of δ.

The Computational Complexity of Coin Flipping. The goal of this paper is to explore the implications of the existence of such (weak[3]) coin flipping protocols for complexity theory. Despite the centrality of randomness and coin flipping to complexity theory and cryptography, surprisingly little is known about this question, as was recently exposited by Impagliazzo[Imp09].

To the best of our knowledge, the only nontrivial results on the subject show, informally, that if one-way functions do not exist, then it must be possible to bias every r-round coin flipping protocol by an additive $\Theta(1/\sqrt{r})$ factor [IL89, CI93, Imp10]. Informally speaking, this does show that $(1 -$ *negligible*$)$-secure weak coin flipping implies the existence of one-way functions. (This result is "tight" for this setting, since if one-way functions exist then weak coin flipping protocols do exist that rule out non-negligible additive bias [Blu82, GL89].)

However, what about other natural notions of significant gain? For example, what are the consequences of $(1-\varepsilon)$-*secure weak coin flipping protocols* – protocols that do not allow a bias of any additive constant ε or more?[4] What about c-secure weak coin flipping protocols where $c \in (0, 1)$ is a fixed constant?

For both these questions, the only consequences known are of the flavor that $\mathsf{PSPACE} \nsubseteq \mathsf{BPP}$. Indeed, it is not difficult to see that if $\mathsf{PSPACE} \subseteq \mathsf{BPP}$, then for any coin flipping protocol, either a cheating Alice could force the output 1 with probability 1 or Bob could force the output 0 with probability 1. This attack would proceed by using the power of PSPACE to perform an iterated min-max (actually max-average) computation, with polynomial look-ahead depth for each round of the protocol. The question before us is whether a similar attack (but with relaxed success goals) could be carried out with much less computational power, for instance with only a constant level of look-ahead, or using a max (instead of max-average) computation – something that intuitively can be carried out with only the power of NP – even though the overall protocol can have polynomially many rounds?

[2] This is also closely related to the notion of "coin flipping with abort", where the requirement, informally speaking, is that unless a cheating party aborts or is "caught cheating" by the honest party, it cannot bias the coin in either direction. Note that such a protocol immediately implies weak coin flipping, since we can define the output of the protocol to be Tails if Alice aborts or is caught cheating, and similarly Heads if Bob aborts or is caught cheating.

[3] Since strong coin flipping and "coin flipping with abort" both imply weak coin flipping, our results of course also apply to the existence of these other types of protocols.

[4] More precisely, for every constant $\varepsilon > 0$, for large enough security parameters 1^k provided as common input to both Alice and Bob, the protocol should not allow additive bias of at least ε.

Our Main Result and Intuition. In this chapter, we show (as our main result) that the existence of any $(3/4 + \varepsilon)$-secure weak coin flipping protocol implies that $\mathsf{NP} \not\subseteq \mathsf{BPP}$. This resolves an open question posed by Impagliazzo [Imp09]: whether $(1-\varepsilon)$-secure weak coin flipping protocols are possible if $\mathsf{P} = \mathsf{NP}$ (we show they are not). To prove this result, we introduce a new attack strategy that we call *Hedged Greedy* that is fundamentally different from previous attacks in this setting.

At an intuitive level, previous attacks [IL89, CI93] work by having the attacking party behave honestly until it notices that it has reached a node where its choice will have a significant effect on the expected outcome assuming honest behavior (conditioned on its choices so far) from that point onwards. The attack only deviates from honest behavior at this one point, and the non-triviality of this attack follows from an argument that there must be at least one round in the protocol where the attacker's choice influences the outcome by an additive factor of at least $\Theta(1/\sqrt{r})$ for an r-round protocol. Informally speaking, as pointed out to us by Impagliazzo [Imp10], this technique fundamentally cannot get a stronger result since a stronger result by this attack would also imply better unconditional attacks on strong coin flipping protocols by fail-stop adversaries, where a bias of $\Theta(1/\sqrt{r})$ is known to be tight (see e.g. [MNS09]).

A conceptually even simpler attack strategy is a "greedy" strategy. To illustrate, let's consider a toy protocol, as described in the protocol tree drawn in Example 1 below.

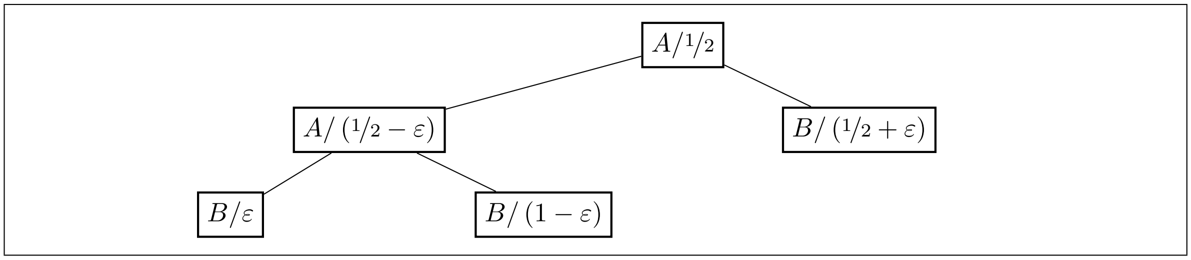

Figure 5.1: Example 1: Motivating Hedged-Greedy.

In this protocol tree, for instance the annotation "$A/1/2$" on the root node denotes two facts: (1) the *value* (or "color") of this node is $1/2$, meaning that an honest execution of the protocol from this node would lead to a coin with expected value $1/2$ (i.e. a fair coin), and (2) the first message of the protocol is sent by Alice, and the bit that is sent determines which child of this node corresponds to the next step of the protocol. The honest Alice will place appropriate probabilities on its children so as to maintain the value of the coin – in this example at the root node, the honest Alice would proceed left with probability $1/2$, and proceed right otherwise. A leaf node marked "B/ε", for instance, just means that honest Bob declares the output to be 1 with probability ε, and declares the output to be 0 otherwise (but Bob has full control over the outcome of the protocol if this leaf node is reached).

We would define the "greedy" attack strategy for Alice (when she is attempting to bias the outcome towards 1) to be one where she always proceeds toward the child with higher value. As this example illustrates, however, this attack may obtain only a tiny additive bias: Here, greedy Alice would proceed to the right child at the root of the tree, obtaining an additive bias of only ε, which can be arbitrarily small. (In this example, however, greedy Bob would still be quite successful. In Appendix B.1, we show an extension of this example where greedy strategies for both Alice and Bob perform poorly.)

In this example, the optimal strategy for Alice is in fact to always proceed to the left child; however, in more complex versions of this example (where "dummy" rounds are added in between the actual rounds of the protocol), it may be very difficult for an attacking Alice to realize that she would have near complete control of the outcome on the left branch of the tree. Given only a bounded look-ahead capability to observe nodes in the protocol tree, for instance, Alice would not be able to ascertain whether or not she can control the output of the protocol on the left due to "dummy" rounds.

Instead, the basic intuitive idea behind our attack strategy is to implement a *hedged greedy* strategy: instead of always following the greedy strategy, our attack will "hedge its bet" by also proceeding to the other child with some probability. Intuitively, when the advantage of behaving greedily is clearer, the attack will potentially deviate more from honest behavior. In the example above, at the root node, since the values of the children are so close, the hedged greedy Alice strategy will place almost equal probabilities to both children (behaving very much like the honest Alice would). But at the level below, on the left, where the values of the two children are so different, the hedged greedy Alice strategy would proceed to the node with value $(1-\varepsilon)$ with probability very close to 1 (thus deviating very strongly from how honest Alice would behave at this node).

In the example above, the hedged greedy Alice strategy would be able to bias the coin to nearly $3/4$. Through a careful choice of the exact hedging behavior of our strategy, we show that in fact we can guarantee similar performance for *any* protocol – in the sense that either hedged greedy Alice will be able to bias to at least roughly $3/4$ or hedged greedy Bob will be able to bias to at most roughly $1/4$. In fact we prove a more general tradeoff between the relative success of hedged greedy Alice and hedged greedy Bob: for example, if hedged greedy Bob *does not* significantly bias the outcome below $1/2$, then we show that hedged greedy Alice must be able to bias the outcome all the way to roughly 1. (In the example above, however, note that even greedy Bob would be able to guarantee the output 0. As such, the example above only illustrates the idea behind our attack, not the actual analysis of it, which is fairly delicate. We are not aware of any simpler attack and analysis that even guarantees a tiny constant additive bias.) We also show that our analysis of our attack is tight, by showing that in fact *any* attack that bases its decisions on only a bounded look-ahead view of the protocol tree (including the values of the nodes, as illustrated above) cannot obtain better bias.

At a technical level, our proof proceeds in two stages: First, we use the power of NP to convert an arbitrary coin flipping protocol to a *stateless* coin flipping protocol with nearly identical security guarantees. In a stateless protocol, honest parties only need the transcript of the protocol so far (and fresh randomness) to determine their next move. We then show an unconditional polynomial-time "hedged greedy" attack on any stateless protocol. We believe this modular approach may be of independent interest.

A stronger result for constant-round protocols. For constant round protocols, it is not difficult to see that the "PSPACE" attack described above can be implemented if $\mathsf{NP} \subseteq \mathsf{BPP}$, since a constant round protocol would only involve a constant number of alternations. A natural question is whether any potentially stronger consequence is true.

For this setting, informally speaking, we obtain essentially the best possible result: even the existence of ε-secure weak coin flipping protocols implies the existence of (infinitely often) one-way functions. The core difference between the setting of $\mathsf{NP} \subseteq \mathsf{BPP}$ and the non-existence of one-way functions is that in the latter case, one only obtains an inverse sampler and approximator that works with high probability over a fixed distribution of inputs. Our attack works by showing how to combine a constant number of inverters for a constant number of related functions to carry out the desired attack.

Conclusions and Future Directions. This work revisits the fundamental question of the computational complexity implications of the existence of coin flipping protocols, where surprisingly little was known. We provide new results which show that in many natural settings of parameters, weak coin flipping protocols in fact imply $\mathsf{NP} \not\subseteq \mathsf{BPP}$, where previously only $\mathsf{PSPACE} \not\subseteq \mathsf{BPP}$ was known. We do this by introducing new techniques for this setting, including a *hedged greedy* attack strategy and a method for its analysis.

A number of important natural questions remain open: For the parameters that we consider in our main result, can we conclude that one-way functions exist (and not just that $\mathsf{NP} \not\subseteq \mathsf{BPP}$)? Is it possible that an ε-secure weak coin flipping protocol (with polynomially many rounds) can exist even if $\mathsf{P} = \mathsf{NP}$? Recently, [HO11] have shown that existence of $\Theta(1)$-secure strong coin tossing protocols imply the existence of one-way functions. Due the a classical reduction of optimal quantum strong coin-tossing [Kit03] to optimal quantum weak coin-tossing [Moc07] by [CK09], we know that extremely good attacks against strong coin-tossing protocols translate into good attacks against weak coin-tossing protocols. The result of [HO11] does not guarantee sufficient bias so that one could conclude that $\Theta(1)$-secure weak coin-tossing protocols also imply existence of one-way functions. Thus, this problem still remains open.

5.1 Preliminaries and Conventions

Consider any 2-party protocol π. We view the transcript generation procedure as traversal of a tree, called the *transcript tree* of π. Any transcript prefix v is a node in this tree and the two extensions of the transcript $v0$ and $v1$ are its two children in the tree. The leaves of the tree are labeled with an output 0 or 1. The depth of the tree D is the communication complexity (maximum number of bits exchanged) of the protocol. Each node is annotated as an Alice node or a Bob node, indicating which party must send the next message in the protocol. When a polynomial blow-up in the round complexity of the protocol is not important, we may consider the Alice and Bob nodes as alternating in any path in the tree. (In Section 5.4.1, where the number of rounds is important, we remove the restriction that the tree is binary, but will retain the convention that Alice and Bob nodes alternate.)

The protocol is specified by a randomized algorithm f_π which takes as input a transcript prefix v and a private "state" and outputs an updated state and a next bit (or, if v is a complete transcript, produces a deterministic binary output based only on v). A protocol is called *stateless* if the state variable is always empty.

Let χ_v be the probability of the output of the protocol being 1 conditioned on v being a prefix of the final transcript (when both parties honestly follow the protocol). We call this the *color* of the node v. We shall denote the subtree rooted at v by S_v. We will assume that the height D of the protocol π is the security parameter. When we mention that some event occurs with high probability (written as w.h.p.) it implies that the probability of that event is at least $1 - \exp(-\Theta(D + 1/\varepsilon))$.

We define a μ-secure protocol for a χ^*-weak coin as follows:

Definition 1 (μ-secure implementation of χ^*-Weak coin). *For $\chi^* \in [0,1]$ and $\mu \in [0,1]$, let $\chi^+ = 1 - \mu(D)(1-\chi^*)$ and $\chi^- = \mu(D)\chi^*$. A protocol π is said to be a μ-secure implementation of χ^* weak coin-flipping, if the outcome is a χ^*-coin if both parties follow the protocol honestly, and either*

1. (Secure when Alice wants 1 and Bob wants 0) *For any efficient (PPT) adversarial Alice strategy, the expected outcome of the protocol when playing against the honest Bob strategy is no higher than χ^+ and for any efficient Bob strategy, the expected outcome when playing against the honest Alice strategy is no lower than χ^-, or*

2. (Secure when Alice wants 0 and Bob wants 1) *For any efficient (PPT) adversarial Alice strategy, the expected outcome of the protocol when playing against the honest Bob strategy is no lower than χ^- and for any efficient Bob strategy, the expected outcome when playing against the honest Alice strategy is no higher than χ^+.*

With increasing μ, the protocol has a better security guarantee: if $\mu = 1$, then neither party can bias the coin away from χ^* towards their desired outcome. Against an adversary with access to a PSPACE oracle, it is easy to see that any efficient protocol is 0-secure. Our attack in Section 5.2 renders any protocol about $1/2$-secure; for $\chi^* = 1/2$, this means that some party can bias the protocol to about $1/4$ (if its desired outcome is 0) or to about $3/4$ (if its desired outcome is 1). (Our attack in Section 5.4.1 on the other hand renders any protocol μ secure with μ close to 0.)

In Section 5.4.1 we show that if a constant round weak coin-flipping protocol must be secure, then a standard weaker variant of one-way functions, called infinitely-often one-way functions must exist. This variant appears in earlier work like [OW93], but seems to have been named so in [HI08]. A polynomial time computable function $f : \{0,1\}^* \to \{0,1\}^*$ is called an infinitely-often one-way function if for any polynomial p and any PPT adversary A, if for infinitely many values n, $\Pr_{x \leftarrow \{0,1\}^n}[f(A(f(x)) = f(x)] < 1/p(n)$ (where the probability is also over the coins of A). Thus, if f is not an infinitely-often one-way function, then there is a PPT adversary A which *for all but finitely many values of* n has a significant probability of inverting f on random inputs from $\{0,1\}^n$.

5.2 Complexity of Weak Coin-Tossing

In this section we show our main result, that if there is a polynomial time weak coin-tossing protocol, then $\mathsf{NP} \not\subseteq \mathsf{BPP}$. In fact, we show that any weak coin-tossing protocol can be attacked significantly (biasing the outcome by close to $3/4$) by polynomial time adversaries with access to an NP oracle. We arrive at this result in a few steps:

- First, we observe that for any polynomial time protocol π, there exists a *state-less* protocol π' that runs in polynomial time with access to an NP oracle, such that π' is "as secure as" π when considering adversaries with access to NP oracles. (Lemma 21.)
- Next we show that, unconditionally, any state-less protocol for weak coin-flipping can be attacked efficiently, using just the protocol itself as a black-box.

Together, these give an attack on any polynomial time protocol π, wherein the attack will use an NP oracle (which will be used to implement the state-less protocol π' that will be accessed as a black-box by the attack).

The first of these follows rather easily from a result on uniform generation of NP-witnesses given an NP oracle [JVV86, BGP00]. (details of which shall be provided in Section 5.2.1) We remark that while much weaker computational power (namely inverting a one-way function) is enough for carrying out such a reconstruction in the normal course of the protocol, it is much harder to ensure that the reconstruction works with adequate accuracy even when the protocol is under attack and may result in a transcript distribution significantly different from that in the normal execution. However, by [JVV86, BGP00], such reconstruction can be accurately carried out for any transcript history when an NP-oracle is given.

Our main work then is in showing an attack on a state-less protocol. Surprisingly we can do this efficiently using the protocol itself as a black-box, and with no further computational complexity assumption. In Section 5.2.2 we provide an intuition for the way the attack works, assuming we have certain additional oracles related to the protcol. Then in Section 5.3 we present the actual attack, which involves additional checks to make the simpler attack robust, and then replaces the oracles it required by approximate implementations.

5.2.1 Private State is Not Useful

Access to an NP-oracle can be used to reconstruct a correctly distributed random tape for a party in a protocol, using just the public history of the protocol. We remark that while much weaker computational power (namely inverting a one-way function) is enough for carrying out such a reconstruction in the normal course of the protocol, it is much harder to ensure that the reconstruction works with adequate accuracy even when the protocol is under attack and may result in a transcript distribution significantly different from that in the normal execution. However, by a result on uniform generation of NP-witnesses given an NP oracle [JVV86, BGP00], such reconstruction can be accurately carried out for any transcript history when an NP-oracle is given. More formally, we need the following result.

Lemma 21. *For any polynomial time protocol π for χ^* weak coin-flipping that is μ-secure against polynomial time adversaries with access to NP oracles, there is a* state-less *protocol π' that runs in (expected) polynomial time with access to an NP oracle, and is also a χ^* weak coin-flipping that is μ-secure against polynomial time adversaries with access to NP oracles.*

Proof Sketch. We shall in fact show a much more general result here: π can be any arbitrary polynomial time protocol (not necessarily for coin-flipping) and we can consider its behavior against any arbitrary class of adversaries. We shall describe a protocol π' which is executed with the help of an NP oracle, such that any adversary's view when interacting with honest players running π and those running π' are identical (or statistically close, if using a strict polynomial time implementation of π').

To define π' we shall use the result of Bellare et al. [BGP00] that for any NP relation $R(\cdot;\cdot)$, there is an expected polynomial time algorithm $S_R^{3-\texttt{SAT}}$ (with access to the oracle for $3-\texttt{SAT}$) such that given x, it samples an element uniformly from the set $R^{-1}(x) := \{w|R(x;w) = 1\}$, provided that this set is non-empty.

Define $R_A(v; r_A) = 1$ iff v is a prefix of the transcript generated when Alice executes the protocol π with a random tape r_A and receives responses from Bob consistent with v. Note that this is indeed an NP relation[5] since Alice's program is polynomial time. Hence, by [BGP00], $S_{R_A}^{3-\mathtt{SAT}}(v)$ outputs a random tape r_A uniformly from $R_A^{-1}(v)$. Similarly define R_B and S_{R_B}.

Define π' to be the stateless protocol in which Alice and Bob behave as follows: on being given a transcript prefix v, Alice uses $S_{R_A}^{3-\mathtt{SAT}}(v)$ to sample a random tape r_A, internally simulates the stateful protocol π with this random tape, and responses from Bob as given in v, and outputs the next bit after v in the transcript so generated. Bob behaves symmetrically, using S_{R_B} instead.

Then, the protocol tree defined for π' is identical to that of π, and for any adversary, the view on attacking π' is the same as that of attacking π.

□

5.2.2 A Simplified Sketch of the Attack

In this section we describe an attack on any weak coin flipping protocol π, given an oracle that, for any partial transcript v, can return χ_v, the color of v in the protocol π.

Given such an oracle for the colors, we define four attacks, two each for corrupt Alice and corrupt Bob — for each party, one to bias the outcome towards 0 and one to bias it towards 1. In Figure 5.2, we describe the attack for Alice to bias the outcome towards 1; the other attacks are symmetric.

Intuition of Attack $\mathsf{Adv}_A^{(1)}$

A D round protocol π with for a χ^*-coin is given. We have access to an oracle which provides the exact color χ_v of any node v.

Suppose the protocol is currently at an Alice-node v (i.e., the next message is sent by Alice). Let $v0$ and $v1$ be its two children. For convenience we write χ, χ_0 and χ_1 respectively for χ_v, χ_{v0}, and χ_{v1}. Let $p_0 = \Pr_\pi[v0|v]$ and $p_1 = \Pr_\pi[v1|v]$ so that $p_0 + p_1 = 1$ and $\chi = p_0\chi_0 + p_1\chi_1$. Given χ, χ_0 and χ_1 we can calculate p_0 and p_1.

- Let $t_b = \frac{p_b\chi_b\left(1-\chi_{(1-b)}\right)}{(\chi-\chi_0\chi_1)}$, for $b \in \{0, 1\}$. Send 0 as the next message with probability t_0 and 1 as the next message with probability t_1.

Figure 5.2: Intuition of Attack $\mathsf{Adv}_A^{(1)}$ for Alice to bias towards outcome 1.

Note that indeed $t_0 + t_1 = \frac{(p_0\chi_0+p_1\chi_1)-(p_0+p_1)\chi_0\chi_1}{\chi-\chi_0\chi_1} = 1$, since $p_0\chi_0 + p_1\chi_1 = \chi$ and $p_0 + p_1 = 1$.

[5] More formally we can include the security parameter on both arguments to R to ensure that it is polynomially balanced.

We shall show that our choice of the probabilities t_0 and t_1 are such that no matter what the protocol is, these attacks break the security of the protocol. More precisely, if we denote the four attacks by $\mathsf{Adv}_A^{(0)}$, $\mathsf{Adv}_A^{(1)}$, $\mathsf{Adv}_B^{(0)}$ and $\mathsf{Adv}_B^{(1)}$ (with $\mathsf{Adv}_A^{(0)}$ corresponding to Alice trying to bias towards 0 and so on), we show that in any such protocol either $\mathsf{Adv}_A^{(1)}$ biases the outcome to 1 with probability at least 0.75, or $\mathsf{Adv}_B^{(0)}$ biases the outcome to 0 with probability at least 0.75. Also, either $\mathsf{Adv}_A^{(0)}$ biases the outcome to 0 or $\mathsf{Adv}_B^{(1)}$ biases the outcome to 1 with probability at least 0.75. Then, the protocol π is not a secure weak coin-flipping protocol.

In order to analyze these attacks, we will define the following functions to assign a score for the (failure of) performance in biasing the orginal color χ at a node to a value x when the goal is to bias towards a bit b: $s_b(x, \chi) := \frac{|b-x|}{|b-\chi|}$ (for $\chi \neq b$). That is,

$$s_1(x, \chi) = \frac{1-x}{(1-\chi)} \qquad\qquad s_0(x, \chi) = \frac{x}{\chi}$$

In addition, we define $s_0(0, 0) := 0$ and $s_1(1, 1) := 0$. Note that the lower the score, the better the performance in biasing towards b.

For any node v in the transcript tree, let $A^{(0)}(v), A^{(1)}(v), B^{(0)}(v), B^{(1)}(v)$ denote the colors induced at the node v by our four attacks. Then, we will show that:

$$A^{(1)}(v), B^{(1)}(v) \in [\chi_v, 1] \text{ and } A^{(0)}(v), B^{(0)}(v) \in [0, \chi_v] \tag{5.1}$$

$$s_1(A^{(1)}(v), \chi_v) + s_0(B^{(0)}(v), \chi_v) \leq 1 \tag{5.2}$$

$$s_0(A^{(0)}(v), \chi_v) + s_1(B^{(1)}(v), \chi_v) \leq 1 \tag{5.3}$$

Intuitively these two inequalities state that if Alice fails to bias the output to b by a significant amount then Bob can bias the output to $(1-b)$ by a significant amount. More precisely, when v is the root of a protocol that yields a fair coin under honest execution ($\chi_v = 1/2$), the first equation above shows that either $s_1(A^{(1)}(v), \chi_v) \geq 1/2$ (which implies that $A^{(1)}(v) \geq 3/4$) or $s_0(B^{(0)}(v), \chi_v) \geq 1/2$ (which implies that $B^{(0)}(v) \leq 1/4$). That is, the protocol is not secure against an Alice who prefers 1 and a Bob who prefers 0. Similarly, the second equation shows that protocol is not secure when Alice and Bob prefer 0 and 1 respectively either.

We will prove the result by induction on the height h of S_v the subtree rooted at v. If $h = 1$, it is trivial to see that both the conditions are satisfied. Let the four tuple associated with the performance of our attack on the S_{vb} be $(A_b^{(0)}, A_b^{(1)}, B_b^{(0)}, B_b^{(1)}) = (A^{(0)}(vb), A^{(1)}(vb), B^{(0)}(vb), B^{(1)}(vb))$. We will only show how the induction works for the first case, i.e. $s_1(A^{(1)}(v), \chi_v) + s_0(B^{(0)}(v), \chi_v) \leq 1$. By induction hypothesis, we know that: $B_b^{(0)} \leq \chi_b$ and

$$\frac{1 - A_b^{(1)}}{(1 - \chi_b)} + \frac{B_b^{(0)}}{\chi_b} \leq 1 \implies (1 - A_b^{(1)}) \leq \left(1 - \frac{B_b^{(0)}}{\chi_b}\right)(1 - \chi_b).$$

(This inequality in fact holds for the extreme cases of $\chi_0 = 0$ and $\chi_0 = 1$ as well: when $\chi_0 = 1$, we have $A_1^{(0)} \in [\chi_0, 1] \implies A_1^{(0)} = 1$; when $\chi_0 = 0$, then $B_0^{(0)} = 0$ and our convention for the score will interpret $\frac{B_0^{(0)}}{\chi_0}$ as 0.)

Suppose v is an Alice node and she outputs b as the next message with probability t_b, where $b \in \{0, 1\}$. Then $A^{(1)}(v) = t_0 A_0^{(1)} + t_1 A_1^{(1)}$ and $B^{(0)}(v) = p_0 B_0^{(0)} + p_1 B_1^{(0)}$.

$$\begin{aligned} s_1(A^{(1)}(v), \chi) + s_0(B^{(0)}(v), \chi) = \frac{1 - A^{(1)}(v)}{(1 - \chi)} + \frac{B^{(0)}(v)}{\chi} &= \frac{t_0(1 - A_0^{(1)}) + t_1(1 - A_1^{(1)})}{(1 - \chi)} + \frac{p_0 B_0^{(0)} + p_1 B_1^{(0)}}{\chi} \\ &\leq B_0^{(0)} T_0 + B_1^{(0)} T_1 + \frac{t_0(1 - \chi_0) + t_1(1 - \chi_1)}{(1 - \chi)} \end{aligned}$$

where $T_0 = \left[\frac{p_0}{\chi} - \frac{t_0(1-\chi_0)}{(1-\chi)\chi_0}\right]$ and $T_1 = \left[\frac{p_1}{\chi} - \frac{t_1(1-\chi_1)}{(1-\chi)\chi_1}\right]$. If we show that $T_0 \geq 0$ and $T_1 \geq 0$, then indeed

$$\begin{aligned} s_1(A^{(1)}(v), \chi) + s_0(B^{(0)}(v), \chi) &\leq \chi_0 T_0 + \chi_1 T_1 + \frac{t_0(1 - \chi_0) + t_1(1 - \chi_1)}{(1 - \chi)} \\ &= \frac{p_0 \chi_0 + p_1 \chi_1}{\chi} = 1 \end{aligned}$$

(using the fact that $B_b^{(0)} \leq \chi_b$). Now, substituting $t_0 = \frac{p_0 \chi_0 (1-\chi_1)}{(\chi - \chi_0 \chi_1)}$ and $t_1 = \frac{p_1 \chi_1 (1-\chi_0)}{(\chi - \chi_0 \chi_1)}$, we observe that

$$T_0 = \frac{p_0\left[(1 - \chi)(\chi - \chi_0\chi_1) - \chi(1 - \chi_0)(1 - \chi_1)\right]}{\chi(1 - \chi)(\chi - \chi_0\chi_1)} = \frac{p_0(\chi_1 - \chi)(\chi - \chi_0)}{\chi(1 - \chi)(\chi - \chi_0\chi_1)} \geq 0$$

(using the fact that $\min\{\chi_0, \chi_1\} \leq \chi \leq \max\{\chi_0, \chi_1\}$), and similarly $T_1 \geq 0$.

Now we need to show that $A^{(1)}(v) \in [\chi, 1]$ and $B^{(0)}(v) \in [0, \chi]$. Note that if $\chi_0 \geq \chi_1$ then $t_0 \geq p_0$ and if $\chi_1 \geq \chi_0$, then $t_1 \geq p_1$. Hence we have $A^{(1)}(v) = t_0\chi_0 + t_1\chi_1 \geq p_0\chi_0 + p_1\chi_1 = \chi$. Also, since $B_0^{(0)} \leq \chi_0$ and $B_1^{(0)} \leq \chi_1$, we have $B^{(0)}(v) = p_0 B_0^{(0)} + p_1 B_1^{(0)} \leq p_0\chi_0 + p_1\chi_1 = \chi$. This completes the analysis of this simplified attack, which assumes t_0 and t_1 can be computed correctly.

Our actual attack is significantly complicated than the one explained in this section, by the fact that we do not have oracles to find χ_v exactly (even given an NP oracle). In fact, we can only estimate χ_v with a small additive error term. The effect of this error on our attack can be severe when χ is very close to 1 or

$(\chi - \chi_0\chi_1)$ is very small. The actual attack takes care of these special cases separately.

5.2.3 A Step Closer to the Actual Attack

We describe our actual attack against a stateless protocol in Section 5.3. The attack closely follows the intuition above, but significantly differs in the details and the analysis. The differences arise from the fact that the above attack depended on accurately knowing certain ratios, which simply cannot be estimated sufficiently accurately.

As described in Figure 5.3, the attack has two additional checks before carrying out an approximate version of the above attack. Firstly, if the color of the current node is very close to 0 or 1, the attack continues by simply following the protocol honestly (even if it later encounters nodes with different colors). Note that this is done even on reaching a node with the color opposite to what the attack desires. The second check is more subtle, and is designed to handle the technical difficulty in accurately estimating t_0 and t_1 when the denominator is close to 0. In the case of $\mathsf{Adv}_A^{(1)}$ (Alice biasing towards 1), this denominator is $\chi - \chi_0\chi_1$; if we see that χ is close to $\min\{\chi_0, \chi_1\}$, then the current step of the attack is changed to weigh the two children using the contribution (in the honest execution) that they make to the color of the current node, i.e., using probabilities $h_b = {}^{p_b\chi_b}/_{\chi}$ instead of t_b. If these checks pass, then the original attack (but with ratios calculated according to the estimated values) is carried out.

In Section 5.3, first we describe the attack in terms of a couple of oracles; but using the fact that π is a stateless protocol, we show that we can indeed implement statistically close approximations of these oracles, using black-box access to (the next message function of) π. Also, the attack needs estimating various quantities with sufficient accuracy, which also can be carried out with black-box access to π.

In Section 5.3.1, we prove the following theorem.

Theorem 8. *Let π be a D round stateless coin-flipping protocol with expected outcome (under honest execution) $\chi \in (0,1)$. For any function (of the security parameter) $0 < \varepsilon < 1$ define $\chi^- = \chi - \frac{\chi}{2}(1-\varepsilon)$, and $\chi^+ = \chi + \frac{(1-\chi)}{2}(1-\varepsilon)$. Then there exist attacks $\mathsf{Adv}_A^{(0)}$, $\mathsf{Adv}_A^{(1)}$, $\mathsf{Adv}_B^{(0)}$ and $\mathsf{Adv}_B^{(1)}$ which use black-box access to π and run in poly$({}^1/_\varepsilon + D)$ time, such that*

1. *$A^{(1)} \geq \chi^+$ or $B^{(0)} \leq \chi^-$, (i.e., not secure if Alice wants 1 and Bob wants 0)*

2. *and, $B^{(1)} \geq \chi^+$ or $A^{(0)} \leq \chi^-$ (i.e., not secure if Bob wants 1 and Alice wants 0).*

where $A^{(b)}$ (resp. $B^{(b)}$), for $b \in \{0,1\}$, is the expectation of the outcome when Alice runs the attack $\mathsf{Adv}_A^{(b)}$ against honest Bob (resp. Bob runs $\mathsf{Adv}_B^{(b)}$ against honest Alice).

As a corollary of Theorem 8, we can conclude that:

Corollary 9. *If a stateless protocol π is a μ-secure polynomial time implementation of χ^*-weak coin for any constant $0 < \chi^* < 1$, then $\mu \le 1/2 + negl(D)$, where D is the number of rounds in π and negl is a negligible function.*

Note that an ideal secure protocol for weak coin-flipping would be a 1-secure protocol. Relative to adversaries with access to a PSPACE oracle, all protocols are 0-secure protocols. Our attack is not as effective as an attack with a PSPACE oracle, but instead renders any (stateless) protocol at most $1/2 + \mathrm{negl}(D)$-secure.

Theorem 8 is proven in Section 5.3.1. Here we give a brief summary. At the heart of the analysis is an analogue of the inductively maintained inequalities equation (5.1)-equation (5.3). These inequalities had depended on the fact the we could arrange the quantities T_0 and T_1 to be positive. Unfortunately, this is no more the case in the analysis of the actual attack. But by carefully choosing our parameters, we can carry out a case analysis and prove Lemma 22. Note that we described the attack assuming access to oracles Π_H and Π_T in addition to Π, and required estimates of various quantities. To complete the proof we show how to estimate these values required by our attack (Lemma 23) and implement (good approximations of) the oracles Π_T and Π_H (Lemma 24) using black-box access to the protocol π. Finally, in Section 5.3.3 we give the choice of parameters to conclude the proof of Theorem 8.

Finally, from Theorem 8 and Lemma 21 we obtain our main result.

Theorem 10. *Let π be a polynomial time (possibly stateful) coin-flipping protocol with expected outcome (under honest execution) $\chi \in (0, 1)$. For any function (of the security parameter) $0 < \varepsilon < 1$ define $\chi^- = \chi - \frac{\chi}{2}(1 - \varepsilon)$, and $\chi^+ = \chi + \frac{(1-\chi)}{2}(1 - \varepsilon)$. Then there exist attacks $\mathsf{Adv}_A^{(0)}$, $\mathsf{Adv}_A^{(1)}$, $\mathsf{Adv}_B^{(0)}$ and $\mathsf{Adv}_B^{(1)}$ which use an NP oracle, but otherwise run in poly$(\frac{1}{\varepsilon} + D)$ time, such that at*

1. *$A^{(1)} \ge \chi^+$ or $B^{(0)} \le \chi^-$,*

2. *and, $B^{(1)} \ge \chi^+$ or $A^{(0)} \le \chi^-$,*

where $A^{(b)}$ (resp. $B^{(b)}$), for $b \in \{0, 1\}$, is the expectation of the outcome when Alice runs the attack $\mathsf{Adv}_A^{(b)}$ against honest Bob (resp. Bob runs $\mathsf{Adv}_B^{(b)}$ against honest Alice).

Proof Sketch. Consider the stateless protocol π' guaranteed by Lemma 21. (This protocol is polynomial time given an NP oracle.) By Theorem 8, there are adversaries $\mathsf{Adv}_A^{(0)}$, $\mathsf{Adv}_B^{(0)}$, $\mathsf{Adv}_A^{(1)}$, $\mathsf{Adv}_B^{(1)}$, which attack π', and access π' as a black-box. Thus these adversaries can be implemented in polynomial time, given an NP oracle. By Lemma 21 (or rather, its proof) these adversaries have the same advantage with π as with π', and hence one of the four conditions stated in Theorem 8 hold with respect to π and these adversaries. Note that these are the same conditions described above, arranged differently. □

Finally, note that if $\mathsf{NP} \subseteq \mathsf{BPP}$, then a highly accurate NP oracle can be implemented in probabilistic polynomial time, and hence the adversaries in the above theorem can be converted to PPT adversaries.

Corollary 11. *If π is a μ-secure polynomial time implementation of χ^*-weak coin for any constant $0 < \chi^* < 1$, then unless $\mathsf{NP} \not\subseteq \mathsf{BPP}$, $\mu \leq 1/2 + negl(D)$, where D is the number of rounds in π and negl is a negligible function.*

In particular, if there is a weak coin-flip protocol for a coin of bias $1/2$ which is μ-secure with $\mu > 1/2 + \alpha$ for some non-negligible function α (corresponding to limiting the bias approximately to the range $[1/4, 3/4]$), then $\mathsf{NP} \not\subseteq \mathsf{BPP}$.

5.3 Attack on Stateless Protocols

In this section we describe our actual attack against a stateless protocol, which follows the intuition of the attack in Section 5.2.2, but does not use exact color oracles. Instead, the attack uses only the protocol itself as a black-box.

Let π be a protocol for coin with bias $\chi^* \in (0, 1)$ (henceforth, called a χ^*-coin). For convenience first we shall describe the attack using a few oracles related to the π:

1. Π: Given a partial transcript v as input, it outputs the next message (bit) as specified by the protocol π. (Recall that the protocol is stateless.)

2. Π_H: Given a partial transcript v as input, it samples a transcript τ as generated by the protocol, conditioned on v being a prefix of τ and the outcome of the protocol at τ being 1 (heads). Then it outputs the bit b such that vb is a prefix of τ.

3. Π_T: Given a partial transcript v as input, it samples a transcript τ as generated by the protocol, conditioned on v being a prefix of τ and the outcome of the protocol at τ being 0 (tails). Then it outputs the bit b such that vb is a prefix of τ.

Attack $\mathsf{Adv}_A^{(1)}$

A D round protocol π with bias χ^* (along with corresponding oracles Π and Π_H) is given. The attack is parametrized by a function of the security parameter $0 < \varepsilon < 1$. Let $\delta = \min\left\{\frac{(1-\chi^*)\varepsilon}{4}, \frac{\chi^*\varepsilon}{4}\right\}$ and $\lambda = \min\left\{\frac{\delta^3}{3^6 D}, \frac{\varepsilon^3\delta^3}{(72)^3 D^3}, \frac{1}{2^9}\right\} = \frac{\varepsilon^3\delta^3}{(72)^3 D^3}$.

Alice performs the following attack at all nodes v where she is supposed to send the next bit. First, compute the following estimates, as described in Lemma 23. Let $\tilde{\chi}$ be an estimate of $\chi := \chi_v$, so that $|\tilde{\chi} - \chi| \leq \lambda$ w.h.p.. Similarly, let $\tilde{\chi}_0$ and $\tilde{\chi}_1$ be estimates of $\chi_0 := \chi_{v0}$ and $\chi_1 := \chi_{v1}$ respectively. Then proceed as follows:

1. If $\tilde{\chi} \geq 1 - (\delta + \lambda)$ or $\tilde{\chi} \leq (\delta + \lambda)$: Henceforth, follow the protocol honestly by making calls to Π. This case takes care of nodes in the transcript tree such that χ is too close to 0 or 1.
2. Else, if $\tilde{\chi} - \min\{\tilde{\chi}_0\,\tilde{\chi}_1\} < \lambda^{1/3} + 2\lambda$, then output $d = \Pi_H(v)$ as the next message. This case takes care of nodes in the transcript tree such that $\chi - \chi_0\chi_1$ is too small.
3. Else (here, $\chi \in [\delta, 1-\delta]$ and $\chi - \min\{\chi_0, \chi_1\} \geq \lambda^{1/3}$), we perform a variant of our original attack. Let $c' \in \{0,1\}$ be such that $\min\{\tilde{\chi}_0, \tilde{\chi}_1\} = \tilde{\chi}_{c'}$ (i.e., the child with lower estimated probability of heads). Let p_0 and p_1 be the probabilities assigned by π to the two possible next messages at v, so that $p_0 + p_1 = 1$ and $\chi = p_0\chi_0 + p_1\chi_1$. Let $\tilde{h}_{c'}$ be an estimation of $h_{c'} = \frac{p_{c'}\chi_{c'}}{\chi}$ such that $\left|\tilde{h}_{c'} - h_{c'}\right| \leq 3\lambda^{1/3}$ (see Lemma 23). Evaluate $\tilde{t}_{c'}$ which is an approximation of

   $$t_{c'} = \frac{p_{c'}\chi_{c'}\left(1 - \chi_{(1-c')}\right)}{(\chi - \chi_0\chi_1)},$$

   such that $\left|\tilde{t}_{c'} - t_c\right| \leq 9\lambda^{1/3}$ (Lemma 23). Set $\tilde{r}_{c'} = \min\{\tilde{t}_{c'}, \max\{0, \tilde{h}_{c'} - 3\lambda^{1/3}\}\}$.

   Send the bit c' with probability $\tilde{r}_{c'}$ and send $1 - c'$ with probability $1 - \tilde{r}_{c'}$.

Figure 5.3: Attack $\mathsf{Adv}_A^{(1)}$ for Alice to bias towards outcome 1.

We will define the four attacks $\mathsf{Adv}_A^{(b)}$ and $\mathsf{Adv}_B^{(b)}$, where $b \in \{0,1\}$ as before: $\mathsf{Adv}_A^{(b)}$ is an algorithm which will provide Alice with a strategy to bias the output towards b. Similarly, $\mathsf{Adv}_B^{(b)}$ will provide a strategy for Bob to bias the output towards b. In Figure 5.3, we explicitly define $\mathsf{Adv}_A^{(1)}$ and all other attacks can be symmetrically defined: algorithm $\mathsf{Adv}_A^{(0)}$ is obtained from $\mathsf{Adv}_A^{(1)}$ by interchanging the interpretations of 1 (Heads) and 0 (Tails). And $\mathsf{Adv}_B^{(b)}$ is defined similarly where that attack is carried out at Bob-nodes in the protocol (i.e., where Bob sends the next bit of the transcript).

The attack refers to oracles Π_H and Π_T, and also estimates of various quantities. But (as we shall see) since π is a stateless protocol, we can indeed implement statistically close approximations of these oracles, and also obtain good estimates of the quantities used in the attack, simply using black-box access to the protocol π.

5.3.1 Proof of Theorem 8

In this section we analyze the protocol in Section 5.3 and prove Theorem 8.

First, similar to the analysis of our simpler attack, we shall prove a lower-bound on the sum of the scores of a pair of attacks.

Lemma 22. *For a stateless weak coin-flipping protocol π, if v is a node in the protocol tree at height h, we have $A^{(0)}(v), B^{(0)}(v) \in [0, \chi_v]$ and $A^{(1)}(v), B^{(1)}(v) \in [\chi_v, 1]$; and*

$$s_1(A^{(1)}(v), \chi_v) + s_0(B^{(0)}(v), \chi_v) \leq 1 + \frac{\delta}{(1-\chi_v)} + \frac{\delta}{\chi_v} + \nu_h$$
$$s_0(A^{(0)}(v), \chi_v) + s_1(B^{(1)}(v), \chi_v) \leq 1 + \frac{\delta}{(1-\chi_v)} + \frac{\delta}{\chi_v} + \nu_h$$

where $A^{(b)}(v)$ (resp. $B^{(b)}$), for $b \in \{0,1\}$, is the expectation of the outcome when Alice runs the attack $\mathsf{Adv}_A^{(b)}$ against honest Bob (resp. Bob runs $\mathsf{Adv}_B^{(b)}$ against honest Alice), and $\nu_0 = 0$ and $\nu_{h+1} = \frac{9\lambda^{1/3}}{\delta} + \nu_h\left(1 + \frac{9\lambda^{1/3}}{\delta}\right)$.

We will just prove the first part of the result, i.e. $s_1(A^{(1)}(v), \chi_v) + s_0(B^{(0)}(v), \chi_v) \leq 1 + \frac{\delta}{(1-\chi_v)} + \frac{\delta}{\chi_v} + \nu_h$. We will proceed by induction on the height of v (i.e., height of S_v, the sub-tree rooted at v). It is easy to see that for the base case of $h = 1$, the result is true since $\chi_v \in \{0,1\}$ and then one of the two terms is 1 and the other is 0 (corresponding to the fact that one of Alice and Bob has zero advantage (and hence score 1) in biasing to their desired value, while for the other party, the outcome is completely biased to their desired value).

Suppose v has height $(h+1)$ and we will use the notation $\chi = \chi_v$, $\chi_0 = \chi_{v0}$ and $\chi_1 = \chi_{v1}$. Let $\chi_{low} = \min\{\chi_0, \chi_1\} = \chi_c$ and $\chi_{high} = \max\{\chi_0, \chi_1\} = \chi_{(1-c)}$, where $c \in \{0,1\}$. If p_0 and p_1 are the probabilities that the next bit after v is 0 and 1, respectively, then we can express $\chi = p_c\chi_{low} + p_{(1-c)}\chi_{high}$. The four tuple summarizing the performance of our attack on the protocol on S_{vb} be $(A_b^{(0)}, A_b^{(1)}, B_b^{(0)}, B_b^{(1)}) = (A^{(0)}(vb), A^{(1)}(vb), B^{(0)}(vb), B^{(1)}(vb))$. By induction hypothesis, we have the following constraint:

$$\begin{aligned} s_1(A_b^{(1)}, \chi_b) + s_b(B_b^{(0)}, \chi_b) = \frac{1 - A_b^{(1)}}{(1-\chi_b)} + \frac{B_b^{(0)}}{\chi_b} &\leq 1 + \frac{\delta}{(1-\chi_b)} + \frac{\delta}{\chi_b} + \nu_h \\ \implies (1 - A_b^{(1)}) &\leq \left[1 + \frac{\delta}{(1-\chi_b)} + \frac{\delta}{\chi_b} + \nu_h - \frac{B_b^{(0)}}{\chi_b}\right](1-\chi_b) \end{aligned}$$

Suppose, Alice is expected to send the bit after v is generated in the protocol. She decides to send 0 with probability $\tilde{r}_0$ and 1 with probability $\tilde{r}_1 = 1 - \tilde{r}_0$. Now, $A^{(1)}(v) = \tilde{r}_0 A_0^{(1)} + \tilde{r}_1 A_1^{(1)}$ and $B^{(0)}(v) = p_0 B_0^{(0)} + p_1 B_1^{(1)}$. We define the following quantity E:

$$\begin{aligned}
E = s_1(A^{(1)}(v), \chi) + s_0(B^{(0)}(v), \chi) &= \frac{1 - A^{(1)}(v)}{(1-\chi)} + \frac{B^{(0)}(v)}{\chi} \\
&= \frac{\tilde{r}_0(1 - A_0^{(1)}) + \tilde{r}_1(1 - A_1^{(1)})}{(1-\chi)} + \frac{p_0 B_0^{(0)} + p_1 B_1^{(0)}}{\chi} \\
&\leq B_0^{(0)} \left[\frac{p_0}{\chi} - \frac{\tilde{r}_0(1-\chi_0)}{(1-\chi)\chi_0}\right] + B_1^{(0)} \left[\frac{p_1}{\chi} - \frac{\tilde{r}_1(1-\chi_1)}{(1-\chi)\chi_1}\right] + \frac{\tilde{r}_0(1-\chi_0) + \tilde{r}_1(1-\chi_1)}{(1-\chi)} \\
&\quad + \frac{(\tilde{r}_0 + \tilde{r}_1)\delta}{(1-\chi)} + \frac{\delta}{(1-\chi)} \left(\frac{\tilde{r}_0(1-\chi_0)}{\chi_0} + \frac{\tilde{r}_1(1-\chi_1)}{\chi_1}\right) + \nu_h \left(\frac{\tilde{r}_0(1-\chi_0) + \tilde{r}_1(1-\chi_1)}{(1-\chi)}\right)
\end{aligned}$$

Let $\tilde{T}_0 = \left[\frac{p_0}{\chi} - \frac{\tilde{r}_0(1-\chi_0)}{(1-\chi)\chi_0}\right]$ and $\tilde{T}_1 = \left[\frac{p_1}{\chi} - \frac{\tilde{r}_1(1-\chi_1)}{(1-\chi)\chi_1}\right]$. Let $\frac{\delta}{\chi'} = \frac{\delta}{(1-\chi)}\left(\frac{\tilde{r}_0(1-\chi_0)}{\chi_0} + \frac{\tilde{r}_1(1-\chi_1)}{\chi_1}\right)$ and $\nu'_h = \nu_h\left(\frac{\tilde{r}_0(1-\chi_0)+\tilde{r}_1(1-\chi_1)}{(1-\chi)}\right)$. We define r_b^* as the value of $\tilde{r}_b$ such that $\tilde{T}_b = 0$. It is impossible to have $\tilde{T}_0, \tilde{T}_1 < 0$ because $r_0^* + r_1^* > 1$ (Lemma 39). So, there are only three cases to consider:

1. If $\tilde{T}_0 \geq 0$ and $\tilde{T}_1 \geq 0$, then

$$\begin{aligned}
E &\leq \chi_0 \tilde{T}_0 + \chi_1 \tilde{T}_1 + \frac{\tilde{r}_0(1-\chi_0) + \tilde{r}_1(1-\chi_1)}{(1-\chi)} + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \nu'_h \\
&= 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \nu'_h = E^{(+,+)}
\end{aligned}$$

2. If $\tilde{T}_0 \geq 0$ and $\tilde{T}_1 < 0$, then:

$$\begin{aligned}
E &\leq \chi_0 \tilde{T}_0 + 0 \cdot \tilde{T}_1 + \frac{\tilde{r}_0(1-\chi_0) + \tilde{r}_1(1-\chi_1)}{(1-\chi)} + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \nu'_h \\
&= \frac{p_0 \chi_0}{\chi} + \frac{\tilde{r}_1(1-\chi_1)}{(1-\chi)} + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \nu'_h \\
&= 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \left(\frac{\tilde{r}_1(1-\chi_1)}{(1-\chi)} - \frac{p_1\chi_1}{\chi}\right) + \nu'_h \\
&= E^{(+,-)}
\end{aligned}$$

3. Similarly, if $\tilde{T}_0 < 0$ and $\tilde{T}_1 \geq 0$, then:

$$\begin{aligned}
E &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \left(\frac{\tilde{r}_0(1-\chi_0)}{(1-\chi)} - \frac{p_0\chi_0}{\chi}\right) + \nu'_h \\
&= E^{(-,+)}
\end{aligned}$$

In our attacks, we intend to use $\tilde{r}_c \leq h_c = \frac{p_c\chi_c}{\chi}$ because:

$$\begin{aligned}
\frac{1}{\chi'} = \frac{1}{(1-\chi)}\left(\frac{\tilde{r}_0(1-\chi_0)}{\chi_0} + \frac{\tilde{r}_1(1-\chi_1)}{\chi_1}\right) &\leq \frac{1}{\chi} \\
\iff \qquad \frac{\tilde{r}_0}{\chi_0} + \frac{\tilde{r}_1}{\chi_1} &\leq \frac{1}{\chi} \\
\iff \qquad \tilde{r}_c\left(\frac{1}{\chi_c} - \frac{1}{\chi_{1-c}}\right) &\leq \left(\frac{1}{\chi} - \frac{1}{\chi_{1-c}}\right) \\
\iff \qquad \tilde{r}_c &\leq \frac{p_c\chi_c}{\chi} = h_c
\end{aligned}$$

The three cases of our attack are analyzed below:

Case 0: Suppose $\tilde{\chi} \geq 1-(\delta+\lambda)$ or $\tilde{\chi} \leq (\delta+\lambda)$, then we know that w.h.p. $\chi \geq 1-(\delta+2\lambda)$ or $\chi \leq (\delta+2\lambda)$. In this case Lemma 41 shows that the induction goes through.

Case 1: In this case $\chi \in [\delta, 1-\delta]$ and $\chi - \chi_c \leq \lambda^{1/3} + 4\lambda$. We use $\tilde{r}_0 = \frac{p_0\chi_0}{\chi}$ and $\tilde{r}_1 = \frac{p_1\chi_1}{\chi}$ and Lemma 42 shows that the induction also works in this case.

Case 2: In this case $\chi \in [\delta, 1-\delta]$ and $\tilde{\chi} - \min\{\tilde{\chi}_0, \tilde{\chi}_1\} \geq \lambda^{1/3} + 2\lambda$. Observe that $|\chi_0 - \chi_1| \geq \lambda^{1/3} \geq 2\lambda$. So, $\tilde{\chi}_0 < \tilde{\chi}_1$ if and only if $\chi_0 \leq \chi_1$. Therefore, $c = c'$.

Since, $(\tilde{\chi} - \min\{\tilde{\chi}_0, \tilde{\chi}_1\}) \geq \lambda^{1/3} + 2\lambda$, we have $(\chi - \chi_0\chi_1) \geq \lambda^{1/3}$. So, we can use Lemma 23 to estimate t_c such that $\left|\tilde{t}_c - t_c\right| \leq 9\lambda^{1/3}$. Recall, $\tilde{r}_c = \min\left\{\tilde{t}_c, \max\left\{0, \tilde{h}_c - 3\lambda^{1/3}\right\}\right\} \leq h_c$ (Lemma 44). Now, Lemma 45 implies that $|\tilde{r}_c - t_c| \leq 9\lambda^{1/3}$ and Lemma 43 shows that the induction works for this case.

Observe that in all cases of our attack, we used $\tilde{r}_c \leq h_c = \frac{p_c\chi_c}{\chi} \leq p_c$ and hence $A^{(1)}(v) = \tilde{r}_c A^{(1)}_c + \tilde{r}_{(1-c)} A^{(1)}_{(1-c)} \geq \tilde{r}_c\chi_c + \tilde{r}_{(1-c)}\chi_{(1-c)} \geq p_c\chi_c + p_{(1-c)}\chi_{(1-c)} = \chi$. And $B^{(0)}(v) = p_0 B^{(0)}_0 + p_1 B^{(0)}_1 \leq p_0\chi_0 + p_1\chi_1 = \chi$. This completes the proof of this lemma.

5.3.2 Estimating Quantities and Implementing Oracles Π_H and Π_T

Estimation of quantities. In our attack, we used estimations of χ, χ_b, p_b, $h_b = \frac{p_b\chi_b}{\chi}$ and $t_b = \frac{p_b\chi_b(1-\chi_{(1-b)})}{(\chi-\chi_0\chi_1)}$. The color of any node can be estimated by sampling N of transcripts which have v as their prefix, according to the honest distribution, and computing the average of all the outcomes. A random transcript can be generated by repeated invocation of the oracle Π. By simple Chernoff Bound, if $N = (D+1/\varepsilon)/\lambda^2$, then the difference between the estimated and actual colors is at most λ with probability $1-\exp(-\Theta(D+1/\varepsilon))$. Similarly, to estimate p_b, invoke Π at v for N times and estimate p_b as the fraction of instances where b is obtained the next bit.

Estimation of h_b and t_b is performed by estimating the individual quantities in the expression and then using them for the calculation. For example, to estimate h_b we first compute $\tilde{p}_b$ (estimation of p_b), $\tilde{\chi}_b$ (estimation of χ_b) and $\tilde{\chi}$ (estimation of χ). Then we define $\tilde{h}_b = \frac{\tilde{p}_b\tilde{\chi}_b}{\tilde{\chi}}$. But the error in estimation could increase significantly if χ is very small. So, this method to estimate h_b should only be used when χ is larger than a particular threshold. Lemma 23 provides the exact details and bounds on these values:

Lemma 23 (Estimation)**.** *In the oracle world we can efficiently find $\tilde{\chi}$, $\tilde{\chi}_0$, $\tilde{\chi}_1$, $\tilde{p}_b$, $\tilde{h}_b$ and $\tilde{t}_b$ such that, w.h.p.:*

1. $\left|\tilde{\chi} - \chi\right|, \left|\tilde{\chi}_0 - \chi_0\right|, \left|\tilde{\chi}_1 - \chi_1\right|, \left|\tilde{p}_c - p_c\right| \leq \lambda$, *and*

2. $\left|\tilde{h}_b - h_b\right| \leq 3\lambda^{1/3}$, *if* $\chi \geq \delta \geq \lambda^{1/3}$ *and* $\lambda \leq 1/3$.

3. $\left|\tilde{t}_b - t_b\right| \leq 9\lambda^{1/3}$, *if* $\chi - \chi_0\chi_1 \geq \lambda^{1/3}$ *and* $\lambda \leq \frac{1}{2^9}$.

Proof. We provide the explicit mechanisms to evaluate these quantities.

1. Estimation of p_b: Call Π N times at node v. Let N_b be the number of times the output of the oracle is b. Define $\tilde{p}_b = N_b/N$. If $N = D/\lambda^2$ then w.h.p. $|\tilde{p}_b - p_b| \leq \lambda$.

2. Estimation of χ, χ_0 and χ_1: Using access to Π, sample N transcripts with prefix v. Let N_1 be the total number of transcripts where the outcome of the coin is 1. Define $\tilde{\chi} = N_1/N$. If $N = D/\lambda^2$ then w.h.p. $|\tilde{\chi} - \chi| \leq \lambda$. Similarly, we can also estimate $\tilde{\chi}_0$ and $\tilde{\chi}_1$.

3. Estimation of h_b: Compute $\tilde{p}_b$, $\tilde{\chi}_b$ and $\tilde{\chi}$ as approximations of p_b, χ_b and χ, such that $|\tilde{p}_b - p_b| \leq \lambda$, $|\tilde{\chi}_b - \chi_b| \leq \lambda$ and $|\tilde{\chi} - \chi| \leq \lambda$. Let $a_1 = p_b\chi_b$. Define the estimation of a_1 as $\tilde{a}_1 = \tilde{p}_b\tilde{\chi}_b$. We know that $|\tilde{a}_1 - a_1| \leq 3\lambda$. Define the estimation of $a_2 = \frac{1}{\chi}$ as $\tilde{a}_2 = \frac{1}{\tilde{\chi}}$. Then, we know that $|\tilde{a}_2 - a_2| \leq \frac{\lambda}{\delta(\delta-\lambda)} \leq 2\lambda/\delta^2 \leq 2\lambda^{1/3}$. Define $\tilde{h}_b = \tilde{a}_1\tilde{a}_2$. Now, $\left|\tilde{h}_b - h_b\right| \leq 2\lambda + 2\lambda^{1/3} \leq 3\lambda^{1/3}$.

4. Estimation of t_b: Suppose we compute $\tilde{\chi}$, $\tilde{\chi}_0$, $\tilde{\chi}_1$ and $\tilde{p}_b$ such that $|\tilde{\chi} - \chi| \leq \lambda$, $|\tilde{\chi}_0 - \chi_0| \leq \lambda$, $|\tilde{\chi}_1 - \chi_1| \leq \lambda$ and $|\tilde{p}_b - p_b| \leq \lambda$. First we will estimate $a_1 = p_b\chi_b\left(1 - \chi_{(1-b)}\right)$. Define $\tilde{a}_1 = \tilde{p}_b\tilde{\chi}_b\left(1 - \tilde{\chi}_{(1-b)}\right)$, then $|\tilde{a}_1 - a_1| \leq 7\lambda$. Next, we will estimate $a_2 = (\chi - \chi_0\chi_1)$. Define $\tilde{a}_2 = (\tilde{\chi} - \tilde{\chi}_0\tilde{\chi}_1)$, then $|\tilde{a}_2 - a_2| \leq 4\lambda$. Let $a_3 = 1/a_2$. If we define $\tilde{a}_3 = 1/\tilde{a}_2$, then:

$$\begin{aligned} |\tilde{a}_3 - a_3| &\leq \frac{4\lambda}{\lambda^{1/3}\left(\lambda^{1/3} - 4\lambda\right)} && \left(\because a_2 \geq \lambda^{1/3}\right) \\ &\leq 8\lambda^{1/3} && \left(\because \frac{1}{\left(\lambda^{1/3} - 4\lambda\right)} \leq \frac{2}{\lambda^{1/3}}\right) \end{aligned}$$

Let $\tilde{t}_b = \tilde{a}_1\tilde{a}_3$, then $\left|\tilde{t}_b - t_b\right| \leq 9\lambda^{1/3}$, because $\lambda \leq \frac{1}{2^9}$. □

Implementing the Oracles Π_H and Π_T. We show how to implement these oracles using only black-box access tothe protocol.

Lemma 24. *Given black-box access to the next message function of a stateless protocol π, we can efficiently implement the oracle Π and provide statistically close approximations of Π_H and Π_T on queries v such that $\chi_v \in [\delta, 1-\delta]$.*

Proof. Implementing Π simply involves picking a random string r uniformly at random and returning the bit $f_\pi(v; r)$, where f_π is the next message function of the protocol π.

Suppose we want to implement Π_H for input v, such that $\chi_v \geq \delta$. We generate $(D + 1/\varepsilon)/\delta$ transcripts which are extensions of v. (This is performed by repeatedly calling Π on u_i starting with $u_{|v|} = v$ and $u_{i+1} := \Pi(u_i)$, till getting a complete transcript $\tau = u_D$.)

If there are no transcripts with outcome 1 (Heads) then we return 0 as the bit after v. Otherwise, return the bit after v in the first transcript which has outcome 1 (Heads). Conditioned on there being such a transcript, the bit produced is correctly distributed. On the other hand, the probability that none of the transcripts has outcome 1 is at most $(1-\delta)^{(D+1/\varepsilon)/\delta} \leq \exp(-D - 1/\varepsilon)$. So the probability of generating the output b is exponentially close to h_b. The oracle Π_T is also implemented similarly.

So, by making at most $D(D + 1/\varepsilon)/\delta$ calls to $f(\cdot; \cdot)$ we can implement $\tilde{\Pi}_H$ and $\tilde{\Pi}_T$ that are statistically close to Π_H and Π_T respectively, for all v such that $\chi_v \in [1-\delta, \delta]$. □

5.3.3 Putting Everything Together

Now we can combine Lemma 22, Lemma 23 and Lemma 24 to obtain Theorem 8. When we run our attack in the oracle world, we have $\nu_D = \left(1 + \frac{9\lambda^{1/3}}{\delta}\right)^D - 1 \leq 18D\lambda^{1/3}/\delta \leq \varepsilon/4$, since $9\lambda^{1/3}/\delta \leq 1$ and $\lambda \leq \left(\frac{\varepsilon\delta}{72D}\right)^3$. So, in the oracle world, for the root node of the protocol we have:

$$s_1(A^{(1)}(v), \chi^*) + s_0(B^{(0)}(v), \chi^*) \leq 1 + \frac{\delta}{(1-\chi^*)} + \frac{\delta}{\chi^*} + \frac{\varepsilon}{4} \leq 1 + \frac{3\varepsilon}{4}$$
$$s_0(A^{(0)}(v), \chi^*) + s_1(B^{(1)}(v), \chi^*) \leq 1 + \frac{\delta}{(1-\chi^*)} + \frac{\delta}{\chi^*} + \frac{\varepsilon}{4} \leq 1 + \frac{3\varepsilon}{4}$$

In the oracle world, the oracles are accessed at most $\text{poly}(D + \varepsilon^{-1})$ times. Since the approximate oracles $\tilde{\Pi}_H$ and $\tilde{\Pi}_T$ are statistically close to the respective original oracles, the attack behavior in the real world and the oracle world differ by at most $\varepsilon/4$. So, in the real world, for the root node of the protocol π we have:

$$s_1(A^{(1)}(v),\chi^*) + s_0(B^{(0)}(v),\chi^*) \leq 1+\varepsilon$$
$$s_0(A^{(0)}(v),\chi^*) + s_1(B^{(1)}(v),\chi^*) \leq 1+\varepsilon$$

Let χ^+ and χ^- be such that $s_1(\chi^+,\chi^*) = s_0(\chi^-,\chi^*) = \frac{1}{2}+\frac{\varepsilon}{2}$. Now, Theorem 8 immediately follows.

5.4 Constant Round Weak Coin-Flipping

In this section we show a much stronger intractability implication of a weak coin-flipping protocol with a very weak unbiasability guarantee, if the protocol has only constantly many rounds. Note that we do allow the communication complexity of the protocol to be polynomial. We show the following result:

Theorem 12. *If infinitely-often one-way functions do not exist then for any constant round coin-tossing protocol π, there exist attacks $\mathsf{Adv}_A^{(0)}$, $\mathsf{Adv}_A^{(1)}$, $\mathsf{Adv}_B^{(0)}$ and $\mathsf{Adv}_B^{(1)}$, such that for any $\varepsilon = 1/poly(k)$ $(0<\varepsilon<1)$, the attacks run in polynomial time in k, and for sufficiently large k:*

1. $\min\left\{1-A^{(1)}, B^{(0)}\right\} \leq \varepsilon$, *and*

2. $\min\left\{A^{(0)}, 1-B^{(1)}\right\} \leq \varepsilon$,

where $A^{(b)}$ (resp. $B^{(b)}$) for $b\in\{0,1\}$, is the expectation of the outcome when Alice runs the attack $\mathsf{Adv}_A^{(b)}$ against honest Bob (resp. Bob runs $\mathsf{Adv}_B^{(b)}$ against honest Alice).

In other words, if infinitely-often one-way functions do not exist then with probability close to 1 either Alice can bias the outcome to b or Bob can bias it to $(1-b)$ starting from any transcript prefix v.

The attack has the following intuitive form: use the fact that any polynomial time function can be inverted to implement next message function oracle for the protocol. At any point in the protocol, use this to sample a polynomial-sized sub-tree of the protocol (with the density of children sampled at each node increasing with depth), and run the PSPACE attack on this sampled tree to decide on the next move in the attack. While conceptually simple, this idea runs into two complications.

- At each round, the response from the honest party may not fall within the sub-tree that was sampled for the attack at that round; and as such the original attack computed may have no further relevance, and no use in deciding the response in subsequent rounds. Further, the PSPACE attack involves evaluating a max-average tree, and by sampling it is quite possible to miss the maximum.

- During the attack the distributions on the nodes at each level of the tree can deviate significantly from that under the honest execution, and the next message function oracles need to work well on these distributions. These distributions depend on the behavior of the attack in the previous rounds, which however carries out recursive look-aheads (in implementing the PSPACE attack), and these look-aheads in turn involve accessing the next message function oracles. A simple attempt at implementing the next message function oracles can lead to circularity.

The second issue can be taken care of by carefully defining a family of next message function oracles, which not only depend on what depth in the protocol it is sampling a message for, but also on which iteration in the PSPACE attack it appears in.

The first issue is addressed by the following intuition: even though it is possible for (say) Alice to miss the maximum (i.e., the child where her advantage is maximum) by a large margin when sampling, this means that a random choice should cause Alice to perform badly; hence this node confers advantage to Bob who is trying to bias the coin in the opposite direction. The actual calculations include more details, and are given in Section 5.4.1.

Implementing Inverters. Our attack is based on realizing an efficient algorithm I, called *inverter*, which can efficiently perform the following task: Given a partial transcript v, it outputs a k-bit message m such that $\Pr(m|v)$ is identical to the probability of π generating a transcript vm conditioned on the fact that v is generated as a partial transcript.

Alternately, if we are able to sample uniformly at random from the set of randomness R_v of pairs (r_A, r_B) such that Alice and Bob with local randomness r_A and r_B generate the partial transcript v when running the protocol π, then we can implement I. We shall reverse sample from the set R_v and run the protocol for one more round and obtain a transcript prefix vm.

We will show the following result:

Lemma 25. *If infinitely-often one-way functions do not exist, then, for sufficiently large k, there exists a class $\mathcal{I} = \{\tilde{I}_{i,j} | i \in [D] \text{ and } j \in [i]\}$ of efficient inverters, such that if Alice uses $\tilde{I}_{i,j}$ to invert v at height j when she is attacking the $(D-i+1)$-th round then the behavior of $\mathsf{Adv}_A^{(1)}$ is at most $1/poly(k)$ different from the case when she uses the actual inverter I.*

Let $f(x) = y$ be a polynomial time function and $\mathcal{D}$ be the distribution of $f(x)$ when x is uniformly sampled. If one-way functions do not exist, then there exists an efficient algorithm A such that $A(y)$ is $1/k^c$ close to the uniform distribution when y is sampled according to the distribution $\mathcal{D}$. Note that the guarantee is only for the distribution $\mathcal{D}$ and not for any arbitrary distribution.

So, we can not use this result to directly create an inverter. The main observation is that the set of nodes $Q_{i,j}$ at height j required to invert when Alice is attacking the i-th round could be performed simultaneously. In other words, they only depend on the nodes inverted while attacking $i' > i$ rounds or at higher levels $j' > j$. So, we define the the execution of $\mathsf{Adv}_A^{(1)}$ just before it inverts $Q_{i,j}$ as the function f. Now, the inverter $I_{i,j} = A$ could be used to invert all partial transcripts in $Q_{i,j}$.

It is worth mentioning that the time complexity of $I_{i,j}$ is only guaranteed to be polynomial in the time complexity of all the inverters $\{I_{i',j'} | i' > i \text{ or } j' > j\}$. So, the time complexity of the inverter $I_{1,1}$ turns out to be $k^{\Theta(1)^D}$, which is polynomial if and only if D is constant. Therefore, this approach works only when D is a constant.

5.4.1 Constant Round Weak Coin-Flipping

In this section we will consider protocols whose transcripts are polynomially long but there are only constant number of rounds (i.e., alternations between Alice and Bob while generating the transcript). In general, the transcript tree can be thought of as a depth D (constant) tree with 2^k fan-out at each node, where k is the security parameter.

Recall that $A^{(b)}(v)$ (resp. $B^{(b)}(v)$) represents the expectation of the outcome when Alice (resp. Bob) wants to bias the outcome towards b in the subtree S_v. We will show the following result:

Theorem 5. *If infinitely-often one-way functions do not exist then for any constant round coin-tossing protocol π, there exist attacks $\mathsf{Adv}_A^{(0)}$, $\mathsf{Adv}_A^{(1)}$, $\mathsf{Adv}_B^{(0)}$ and $\mathsf{Adv}_B^{(1)}$, such that for any $\varepsilon = 1/poly(k)$ $(0 < \varepsilon < 1)$, the attacks run in polynomial time in k, and for sufficiently large k:*

1. $\min\left\{1 - A^{(1)}, B^{(0)}\right\} \leq \varepsilon$, *and*

2. $\min\left\{A^{(0)}, 1 - B^{(1)}\right\} \leq \varepsilon$,

where $A^{(b)}$ (resp. $B^{(b)}$) for $b \in \{0,1\}$, is the expectation of the outcome when Alice runs the attack $\mathsf{Adv}_A^{(b)}$ against honest Bob (resp. Bob runs $\mathsf{Adv}_B^{(b)}$ against honest Alice).

In other words, if infinitely-often one-way functions do not exist then with probability close to 1 either Alice can bias the outcome to b or Bob can bias it to $(1 - b)$ starting from any transcript prefix v.

5.4.2 Oracle World

For simplicity, we will prove Theorem 12 in an oracle world where we have access to *inverters*. An inverter I, when presented with a partial transcript v, honestly extends v by one round. The challenge is to implement these inverters so that they work well *on the distributions effected by the attack* (which in turn depends on inverters). In Section 5.4.3, we will show how to efficiently approximate these inverters, if infinitely-often one-way functions do not exist, such that the actual algorithm's execution differ from the execution of the attack in the oracle world by at most $1/\text{poly}(k)$.

Hypothetical Attack. First we present the ideal attack we want to perform when we are provided access to inverters. We will describe the attack for Alice to bias towards 1 and other attacks can be analogously defined. When Alice wants to bias the outcome towards 1, she attacks at all nodes in the transcript tree which are Alice nodes. Suppose v is a partial transcript generated during the execution of the protocol. Our attack is recursively defined. Let h be the height of the node v in the transcript tree and $A^{(1)}(v)$ be the expected outcome when Alice is trying to bias towards 1 by performing her attack v onwards. For a leaf, $A^{(1)}(v)$ is defined to be the color of v and for a Bob node v, $A^{(1)}(v)$ is the expectation of $A^{(1)}(u)$, where u is a honest extension of the partial transcript v. When v is an Alice node, we shall use the following strategy for Alice: Alice will sample N_h extensions of the partial transcript v, i.e. $\{u_1, \ldots, u_{N_h}\}$, and finds $i \in N_h$ such that $A^{(1)}(u_i) = \max_{i \in [N_h]} u_i$. She sends the next message so that the computation moves to the node u_i in the transcript tree. Thus, $A^{(1)}(v)$ is defined as the expectation of $\max_{i \in [N_h]} u_i$ where each u_i is an honest extension of v. The quantity N_h will be defined suitably later in this section. We remark that, for every node v in the transcript tree, $A^{(1)}(v)$ or $B^{(0)}(v)$ are close to 1 or 0 respectively. This statement will be formalized as a lemma later in this section and we will also see how we can choose our parameters so that Alice or Bob can force 1 or 0 with near certainty.

Actual Attack. Despite having access to inverters, it is extremely hard to exactly compute the expected $A^{(1)}(u)$ when u is an honest extension of v. The problem is considerably harder when we try to compute the expectation $\max_{i \in [N_h]} u_i$. Instead, we will try to estimate the performance of the hypothetical attack using repetitive sampling. Note that we might incur an additive error in our estimation and, with extremely low probability, our estimation could be completely wrong. Thus, we will try to compute $\tilde{A}^{(1)}(v)$ and $\tilde{B}^{(0)}(v)$ which are good estimations of $A^{(1)}(v)$ and $B^{(0)}(v)$ respectively with high probability; and we will recursively use them in our attack instead of the exact $A^{(1)}(v)$ and $B^{(0)}(v)$ values.

Formally, the functions $\tilde{A}^{(1)}(v), \tilde{B}^{(0)}(v)$ are such that, with probability $(1 - \varepsilon_h)$, the following conditions are satisfied:

1. $\left| A^{(1)}(v) - \tilde{A}^{(1)}(v) \right| \leq \varepsilon_h$,

2. $\left| B^{(0)}(v) - \tilde{B}^{(0)}(v) \right| \leq \varepsilon_h$, and

3. $\min\left\{1-\tilde{A}^{(1)}(v), \tilde{B}^{(0)}(v)\right\} \leq \varepsilon_h$,

where h is the height of v in the transcript tree, $\varepsilon_{h+1} = \Gamma^{1/2}\varepsilon_h^{1/6}$ and $\varepsilon_0 = \Gamma^{3/5}\varepsilon^{6^{D+1}}$ and Γ is a parameter which will be defined later. The parameter Γ will be a large number which will be of the form $(1/\varepsilon)^{\Theta(1)}$.

Given, a particular $\varepsilon = 1/\mathrm{poly}(k)$, we will show how to perform our attack. We will prove the following result:

Lemma 26. *In the oracle world where we have access to the ideal inverters, we can efficiently implement $\tilde{A}^{(1)}(v)$ and $\tilde{B}^{(0)}(v)$ for all partial transcripts v; and $\min\{1 - A^{(1)}(v), B^{(0)}(v)\} \leq \varepsilon_h$, where h is the height of the node v in the transcript tree.*

We emphasize that the condition $\min\{1-A^{(1)}(v), B^{(0)}(v)\} \leq \varepsilon_h$ is not probabilistic, unlike the properties of the quantities $\tilde{A}^{(1)}(v)$ and $\tilde{B}^{(0)}(v)$. We also do not try to obtain tighter bounds because the qualitative result does not change; although as intermediate steps, we will prove and use tighter bounds on these quantities. To prove this lemma, we will proceed by induction on the height h of the node v. Recall that leaves, which correspond to complete transcripts, have height $h = 0$ and the root r of the transcript tree has height $h = D$. For the base case, consider any node v at height 0, i.e. either Alice or Bob announces that the outcome is 0 or 1. In this case, it is trivial to implement $\tilde{A}^{(1)}(v)$ and $\tilde{B}^{(0)}(v)$.

For the inductive step, we will show that given an implementation of these functions for nodes with height h we can implement these functions for any node at height $(h + 1)$. W.l.o.g., let v be an Alice node at height $(h + 1)$.

Computation of $\tilde{A}^{(1)}(v)$. Prepare a set $\{u_1, \ldots, u_{N_{h+1}}\}$, where $N_{h+1} = \varepsilon_h^{-1/2}$, of honest extensions of the partial transcript v. Find the maximum $\tilde{A}^{(1)}(u_i)$, for $i \in [N_{h+1}]$. Recall, that $A^{(1)}(v)$ is the expected outcome of this experiment. Perform this task $M_{h+1} = \Gamma^{1/3}\varepsilon_h^{-1/3}$ times and define $\tilde{A}^{(1)}(v)$ as the average of the respective maximums. Intuitively, we are repeating the experiment M_{h+1} times to obtain a good estimation of $A^{(1)}(v)$. We will prove that this definition of $\tilde{A}^{(1)}(v)$ suffices by considering several intermediate hybrid worlds.

1. As an intermediate world, suppose we have access to the ideal values of $A^{(1)}(u)$, where u is a honest extension of the transcript v. $A^{(1)}(v)$ is defined as the expected outcome when we honestly sample N_{h+1} children of v and compute their average. Repeating this experiment M_{h+1} times helps us estimate $A^{(1)}(v)$ with ζ such that, with probability $1 - \exp(-\Theta(\Gamma))$, the quantity $\left|\zeta - A^{(1)}(v)\right|$ is at

most $\sqrt{\frac{\Gamma}{M_{h+1}}} = k^{1/3}\varepsilon_h^{1/6}$. We will choose $\Gamma = \Theta\left(\left(\frac{1}{\varepsilon^{6^D}}\right)^{5/8}\right)$ such that,

$$\begin{aligned}\exp(-\Theta(\Gamma)) &\leq \Theta\left(\frac{1}{\Gamma}\right)\\ &\leq \Gamma^{3/5}\varepsilon^{6^D} = \varepsilon_1\\ &\leq \varepsilon_h\end{aligned}$$

This implies that the expression $1 - \exp(-\Theta(\Gamma))$ in the Chernoff bound is at least $1 - \varepsilon_h$, i.e. with probability $(1 - \varepsilon_h)$ our estimation is within $k^{1/3}\varepsilon_h^{1/6}$ additive error of the actual value of $A^{(1)}(v)$.

2. Now, we replace every ideal value of $A^{(1)}(u)$ with $(A^{(1)}(u))'$ such that $(A^{(1)}(u))' = A^{(1)}(u)$ with probability $1-\varepsilon_h$. In this hybrid, we get a new estimate ζ' such that, with probability $1-M_{h+1}N_{h+1}\varepsilon_h - \varepsilon_h = 1 - k^{1/3}\varepsilon_h^{1/6} - \varepsilon_h$, the error in our estimation $\left|\zeta' - A^{(1)}(v)\right|$ is at most $k^{1/3}\varepsilon_h^{1/6}$. This step follows from union bound.

3. Finally, replacing $(A^{(1)}(u))'$ values with the $\tilde{A}^{(1)}(u)$ values, the new estimate $\tilde{A}^{(1)}(v)$ can deviate at most ε_h away from the estimated ζ'. This step follows from the fact that $\tilde{A}^{(1)}(v)$ is a convex linear combination of $\tilde{A}^{(1)}(u)$ values. Thus, we can conclude that $\left|A^{(1)}(v) - \tilde{A}^{(1)}(v)\right| \leq k^{1/3}\varepsilon_h^{1/6} + \varepsilon_h \leq \varepsilon_{h+1}$, with probability at least $1 - k^{1/3}\varepsilon_h^{1/6} - \varepsilon_h \geq 1 - \varepsilon_{h+1}$.

Computation of $\tilde{B}^{(0)}(v)$. Compute $\{u_1, \ldots, u_{M_{h+1}}\}$ honest extensions of the partial transcript v. Define $\tilde{B}^{(0)}(v)$ as the average of $\tilde{B}^{(0)}(u_i)$, where $i \in [M_{h+1}]$. Similar to the argument presented above, with probability at least $(1-M_{h+1}\varepsilon_h-\varepsilon_h) \geq (1-\varepsilon_{h+1})$, the quantity $\left|B^{(0)}(v) - \tilde{B}^{(0)}(v)\right|$ is at most $M_{h+1}\varepsilon_h+\varepsilon_h \leq \Gamma^{1/3}\varepsilon_h^{1/6} + \varepsilon_h \leq \varepsilon_{h+1}$.

Attacks are good. For the final step in our inductive proof, we need to show that the quantities $\tilde{A}^{(1)}(v)$ and $\tilde{B}^{(0)}(v)$ satisfy the third property of our theorem, i.e. at least one them is a very good attack; and we also need to show that $A^{(1)}(v)$ or $B^{(0)}(v)$ is (respectively) close to 1 or 0 as well. Let $(1 - p)$ be the probability of $A^{(1)}(u) \geq 1 - \varepsilon_h$, where u is some honest extension of v by one round. Let $q \geq p$ (since, inductively, $A^{(1)}(u)$ or $B^{(0)}(v)$ is within ε_h of 1 or 0 respectively) be the probability of $B^{(0)}(u) \leq \varepsilon_h$, where u is some honest extension of v by one round. Observe that the maximum of N_{h+1} samples of $A^{(1)}(u)$ is at least $1 - \varepsilon_h$, unless each one of the sampled $A^{(1)}(u)$s were less than $1 - \varepsilon_h$. Thus, the expected outcome $A^{(1)}(v)$ is at least $\left(1 - p^{N_{h+1}}\right)(1 - \varepsilon_h)$. Similarly, one can argue that $B^{(0)}(v) \leq (1 - p) + p\varepsilon_h \leq (1 - p) + \varepsilon_h$. There are two cases to consider:

1. If $p \geq 1-\varepsilon_h^{1/4}$, then $B^{(0)}(v) \leq \varepsilon_h^{1/4}+\varepsilon_h$.

2. If $p \leq 1-\varepsilon_h^{1/4}$, then

$$\begin{aligned}
A^{(1)}(v) &\geq \left(1-p^{N_{h+1}}\right)(1-\varepsilon_h) \\
&\geq \left(1-\left(1-\varepsilon_h^{1/4}\right)^{N_{h+1}}\right)(1-\varepsilon_h) && \text{, since } p \leq 1-\varepsilon_h^{1/4} \\
&\geq \left(1-\exp\left(-N_{h+1}\varepsilon_h^{1/4}\right)\right)(1-\varepsilon_h) && \text{, since } (1-x) \leq \exp(-x) \\
&= \left(1-\exp\left(-\varepsilon_h^{-1/4}\right)\right)(1-\varepsilon_h) && \text{, since } N_{h+1}=\varepsilon_h^{-1/2} \\
&\geq (1-\varepsilon_h^{1/4})(1-\varepsilon_h) && \text{, since } x \leq 1 \implies \exp(-1/x) \leq x \\
&\geq 1-\varepsilon^{1/4}-\varepsilon_h && \text{, by expansion}
\end{aligned}$$

So, we obtain that $\min\{1-A^{(1)}(v), B^{(0)}(v)\} \leq \varepsilon^{1/4}+\varepsilon_h \leq \varepsilon_{h+1}$. Finally, with probability $(1-\varepsilon_{h+1})$, we have $\min\{1-\tilde{A}^{(1)}(v), \tilde{B}^{(0)}(v)\} \leq (\Gamma^{1/3}\varepsilon_h^{1/6}+\varepsilon_h)+(\varepsilon_h^{1/4}+\varepsilon_h) \leq \varepsilon_{h+1}$. This completes the induction step and the proof of the lemma.

Figure 5.4 describes the algorithm to implement $\tilde{A}^{(1)}(v)$. One important observation is that, since the number of rounds D is constant, we have $\varepsilon_i \geq \varepsilon_1 = \Gamma^{3/5}\varepsilon^{\Theta(1)}$. Since, at each round the time complexity of our attack is $\mathrm{poly}(1/\varepsilon_i)$, our attack runs in polynomial time. Moreover, it is also easy to see that $1-A^{(1)}(\mathrm{r})$ or $B^{(0)}(\mathrm{r})$ is at most ε, where r is the root of the transcript tree.

Subroutine to compute $\tilde{A}^{(1)}(v)$

1. If v is a leaf, return $\tilde{A}^{(1)}(v) = \chi_v$.
2. If v is an Alice node at height j: Sample $\{u_1, \dots, u_{N_j M_j}\}$ honest extensions of v by calling $I(v)$. Return
$$\tilde{A}^{(1)}(v) = \frac{\sum_{k=1}^{M_j} \left[\max_{k'=1}^{N_j} \tilde{A}^{(1)}\left(u_{(k-1)N_j+k'}\right)\right]}{M_j}$$
3. If v is a Bob node at height j: Sample $\{u_1, \dots, u_{M_j}\}$ honest extensions of v by calling $I(v)$. Return
$$\tilde{A}^{(1)}(v) = \frac{\sum_{k=1}^{M_j} \tilde{A}^{(1)}(u_k)}{M_j}$$

Algorithm $\mathsf{Adv}_A^{(1)}$

Let v be an Alice node at height j.

1. Sample $\{u_1, \dots, v_{N_j}\}$ honest extensions of v using $I(\cdot)$.
2. Output the message m such that $vm = \text{argmax}_{k \in [N_j]} \tilde{A}^{(1)}(u_k)$.

Figure 5.4: Computation of $\tilde{A}^{(1)}(v)$ to help Alice bias towards outcome 1.

5.4.3 Implementing Efficient Inverters

Before we proceed, let us recall some terminology related to nodes in the transcript tree. We will assume that every complete transcript is comprised of D message exchanges. The root of the transcript tree is at *level* 0; and every other node is at level one more than its ancestor. So, a full transcript describes a traversal of nodes which are respectively at levels $0, 1, \dots, D$. A level i node is reached as a consequence of i message exchanges between Alice and Bob, for $i \in [D] \cup \{0\}$. The *height* of a leaf of the transcript tree is 0; and every internal node has height one more than its child. So, a leaf to root path visits nodes with height $0, 1, \dots, D$ in that particular order. If a node has height h, then there are h more message exchanges between Alice and Bob remaining before the full transcript is generated. Notice that the sum of the height and level of a node is always D. Next, using this terminologies, we shall prove Theorem 12.

Given access to efficient inverters, Section 5.4.2 shows how Alice and Bob can efficiently bias the outcome ε-close to b or $(1 - b)$, respectively. In this section, we will show that if infinitely-often one-way functions do not exist then for sufficiently large Γ, we can efficiently implement close approximations of these inverters.

A closer look at our attack reveals that the inverters are used in the following manner. Suppose Alice wants to bias the outcome to 1. Let the current partial transcript be v (suppose $(D - i)$ message exchanges have already taken place) and Alice is supposed to send the next message of the transcript. To generate her message, she needs to implement $\tilde{A}^{(1)}(\cdot)$ functions for some nodes in S_v. In fact, she samples a subtree of S_v such that, for some node u in it:

1. If u is an Alice node, then its degree is $N_j M_j$, and

2. If u is a Bob node, then its degree is M_j,

where j is the height of u in the transcript tree. So, in other words, to generate the $(D-i+1)$-th message, Alice accesses polynomially many inverters on nodes u which are at height i, $(i-1)$, ..., 1 in that order. Let $\mathcal{Q}_{i,j}$ be the set of nodes at height j queried by Alice. Observe that all queries in $\mathcal{Q}_{i,j}$ are independent of each other; and without loss of generality, we can assume that all queries in $\mathcal{Q}_{i,j}$ are performed simultaneously. The number of queries performed is upper bounded in the following manner: $|\mathcal{Q}_{i,j}| \le \prod_{k=j}^{i} M_k N_k$.

Let $\tilde{I}_{i,j}$ be the inverter used by Alice to query the nodes at height j when she is generating the $(D-i+1)$-th message. We will provide an inductive construction of $\tilde{I}_{i,j}$, where $i \in [D]$ and $j \in [i]$. Define $\mathcal{I}_{i,j}$ as the collection of all inverters of the form $\left\{\tilde{I}_{i',j'} | (D \ge i' > i \text{ or } D \ge j' > j) \text{ and } j' \in [i'] \text{ and } i' \ge i\right\}$. Let $\mathcal{A}(r_A)$ be the attack algorithm for Alice when she tries to bias the outcome to 1, where r_A is its random tape and she uses the oracles provided in $\mathcal{I}_{0,0}$. We can assume, without loss of generality, that the random tape used by $\mathcal{A}(r_A)$ to run an instance of the inverter $\tilde{I}_{i,j}$ is independently chosen. Let $\mathcal{A}^{(\mathrm{pre})}_{i,j}(r_A)$ be the execution of $\mathcal{A}(r_A)$ till it makes the $\mathcal{Q}_{i,j}$ queries and outputs the set $\mathcal{Q}_{i,j}$, using the oracles provided in $\mathcal{I}_{i,j}$. Consider the following machine $\mathcal{C}(k, \mathcal{A}^{(\mathrm{pre})}_{i,j})$: It samples r_A and r_B uniformly at random and simulates a run of $\mathcal{A}^{(\mathrm{pre})}_{i,j}(r_A)$ against a honest Bob with local randomness r_B. Let $\mathcal{C}^*(k, \mathcal{A}^{(\mathrm{pre})}_{i,j})$ be the machine which runs $\mathcal{C}(k, \mathcal{A}^{(\mathrm{pre})}_{i,j})$ and concatenates $r_A \circ r_B$ at the end.

If infinitely-often one-way functions do not exist, then for any constant c and sufficiently large k, there exists an efficient machine $\tilde{I}_{i,j}$ such that [OW93]:

$$\left\| \mathcal{C}^*(k, \mathcal{A}^{(\mathrm{pre})}_{i,j}) - \mathcal{C}(k, \mathcal{A}^{(\mathrm{pre})}_{i,j}) \circ \tilde{I}_{i,j}(\mathcal{C}, \mathcal{A}^{(\mathrm{pre})}_{i,j}, 0^{k^c D^2 |Q_{i,j}|}) \right\|_1 \le \frac{1}{k^c D^2 |Q_{i,j}|}$$

Note that the time complexity of $\tilde{I}_{i,j}$ is at most $k^{\Theta(1)^{D^2}}$, which is a polynomial because D is a constant. Since there are finitely many inverters $\tilde{I}_{i,j}$ and four different attacks, for sufficiently large k all inverters used in each of our attacks perform well. By union bound, the behavior of our attacks when provided with $\mathcal{I}_{0,0}$ instead of the actual inverters $I(\cdot)$ can differ by at most $1/k^c$. This completes the proof of Theorem 12.

Chapter 6

Separations

In our framework, we have seen that all our earlier results showed that reductions of the form $\mathcal{F} \sqsubseteq^{\mathrm{PPT}} \mathcal{G}$ are equivalent to either OWF assumption or $\mathsf{sh\text{-}OT}$ assumption. In this chapter, we will look at reductions such that the computational assumptions necessary and sufficient to securely realize them define potentially new assumptions which are intermediate to OWF assumption and $\mathsf{sh\text{-}OT}$ assumption. We shall consider two-party semi-honest deterministic SSFE.

Two-party deterministic function evaluations which have perfectly secure protocols against semi-honest adversaries were characterized by Kushilevitz and Beaver [Kus89, Bea89]. They showed that decomposable $\mathcal{F}$ (Chapter 2) are the only two-party deterministic functions which have perfectly semi-honest secure protocols. It was shown recently that this characterization also extends when we consider statistical semi-honest security [MPR09]. The functionalities are called semi-honest trivial functionalities. We emphasize that this characterization extends to functionalities where the input domain sizes of the two parties are bounded by a polynomial in the security parameter. Further, there are extremely complex functionalities which need extremely strong computational assumptions to securely realize them against semi-honest adversaries. Kilian [Kil91] showed that if a deterministic SSFE $\mathcal{F}$ has an OR-minor then $\mathcal{F} \sqsubseteq^{\mathrm{PPT}} \emptyset$ is equivalent to $\mathsf{sh\text{-}OT}$ assumption. Moreover, any such $\mathcal{F}$ is also complete, i.e. using $\mathcal{F}$ as a black-box we can securely compute any other functionality against semi-honest adversaries.

When the output space of these functionalities is of size at most 3, then every two-party deterministic SSFE is either semi-honest trivial or complete [CK89, Kre11]. But if the output space is greater than 4, then there are functions which are neither semi-honest trivial nor complete (refer Figure 1.3). Let $\mathcal{C}_{\mathcal{F}}$ be the assumption: "There exists a semi-honest secure protocol for $\mathcal{F}$", where $\mathcal{F}$ is an intermediate deterministic two-party SSFE. It is clear that $\mathcal{C}_{\mathcal{F}}$ is not unconditionally true and it implied by the $\mathsf{sh\text{-}OT}$ assumption. Furthermore, incorporating the techniques of [IL89] in the proof of [MPR09], we can show that $\mathcal{C}_{\mathcal{F}}$ implies OWF assumption. Characterizing the exact complexity of this assumption is the main focus of this chapter. To claim that this assumption is *distinct* from the known assumptions intermediate to OWF assumption and OWF assumption, we need to show that OWF assumption and PKE assumption do not imply $\mathcal{C}_{\mathcal{F}}$.

Additionally, $\mathcal{C}_{\mathcal{F}}$ does not imply the sh-OT assumption. These questions are dealt with in [MMP11] and in this chapter we shall present partial results to this end.

We emphasize that the exact characterizations of complexity classes P and BPP are still unknown and, thus, it seems difficult to define when one cryptographic primitive cannot be constructed given another. Impagliazzo and Rudich [IR89] created a framework to show when one computational assumption does not imply another computational assumption, when we consider some specific types of protocols. For example, they showed a relativized world where OWF assumption holds but KA assumption does not. This proves that any secure KA protocol cannot be based on black-box usage of OWF. In general, any relativizing technique cannot be used for building KA schemes based on OWF assumption. More formally, they rule out fully black-box constructions as defined by [RTV04]. We note that non black-box techniques in theoretical computer science, in general, and cryptography, in particular, are extremely rare. Only recently, have we successfully learned to use the code of the adversary in security reductions [Bar01]. So, such relativized separations are interpreted to imply that the cryptographic primitives capture distinct classes of computational intractability assumptions. It is known that OWF assumption, KA assumption, PKE assumption and sh-OT assumption are all distinct. Following [IR89] many other black-box separation results followed [Sim98, GMR01, BPR^{+}08, KSY11, MM11] (which this list is only a partial incomplete one). Another trend of results is to prove lower-bounds on the efficiency of the implementation reduction in black-box constructions [KST99, GGKT05, LTW05, HHRS07, BMG07, BM09, HHRS07]. A complementary approach has been to find black-box reductions for cases that non-black-box reductions were known originally [GMW91, IL89, Ost91, OW93, Hai08, HNO^{+}09]. Similar results are mentioned in Section 2.1.2.

The technique introduced in [IR89] considers an oracle $\mathcal{O}$ which provides parties access to a random oracle and a PSPACE oracle. Providing access to the PSPACE oracle implies that any non-triviality in security achieved by protocols in this relativized world can be attributed to the random oracle and not on the bounded computational power of the parties. In this relativized world, it is easy to see that one-way functions exist. Even parties with unbounded computational power which perform only polynomially many queries to the random oracle, cannot invert a random image of the random oracle with non-negligible probability. They show that if there exists any secure KA protocol where parties access $\mathcal{O}$ then there exists a KA protocol where parties do not access the random oracle (but still have access to the PSPACE oracle). We know that this is impossible, because any eavesdropper can successfully guess the secret with non-negligible probability if the parties agree on a secret with non-negligible probability. In this chapter we shall show that $\mathcal{C}_{\mathcal{F}}$ is separated from OWF assumption. We will also generalize this result to a wide class of oracles called atomic sub-modular oracle. Further arguments in [MMP11] show that $\mathcal{C}_{\mathcal{F}}$ is also separated from

PKE assumption.

Random oracles have proven to be extremely powerful in cryptography, for example [FS86], to securely realize information theoretically impossible cryptographic constructs and improve round complexity of cryptographic schemes. For example, using a random oracle we can construct information theoretically standalone secure commitment schemes, which are impossible in the plain model. In fact, these schemes can be perfectly binding, with probability close to 1, if the random oracle is length tripling. Although random oracles are used to capture the idealized versions of several properties of cryptographic functions, for example one-way functions etc., a security proof in the random oracle model is not accepted as a convincing proof, following [CGH98]. But, it is still used for heuristic arguments of security and as an idealized version of cryptographic primitives like one-way functions, collision resistant hash functions etc. Similar to [IR89], we show that random oracles are useless for two-party deterministic SSFE.

6.1 Techniques

Our approach is a generalization of the techniques introduced in [BM09]. We will build and improve their approach to show that if parties have unbounded computational power then random oracles are useless for them for semi-honest secure computation. Next, we apply the semi-honest attack on undecomposable SSFE introduced in [MPR09] to this compiled protocol. We emphasize that the approach in [MMP11] uses similar ideas but, instead of generalizing the results of [BM09], reduces the current problem to the results in [BM09, DLMM11].

Compilation. The first step in our proof is to show that if an undecomposable SSFE $\mathcal{F}$ has semi-honest secure protocol when parties have access to $\mathcal{O} = (\mathcal{R}, \mathsf{PSPACE})$, where $\mathcal{R}$ is a random oracle, then there exists a semi-honest secure protocol for $\mathcal{F}$ when parties have access to PSPACE oracle only. This compiled protocol has *similar* completeness and security guarantees. Barak and Mahmoody [BM09] showed a similar result which proved that access to random oracles is useless for key-agreement protocols. The main idea is to create a curious eavesdropper Eve who asks highly likely queries based on her view. The eavesdropper's view comprises of the publicly generated transcript and the prior query-answer list she obtained by querying the random oracle. Their Eve algorithm queried all highly likely queries and when she stopped, the probability of any query being in Alice or Bob view conditioned on her local view was below a threshold ε. Additionally, the joint view of Alice and Bob views consistent with Eve's view was close to a product distribution. Thus, Eve could sample one Bob view consistent with her current view and can predict the common key agreed by Alice and Bob with significant probability.

This strategy does not suffice for our problem. First problem is that Alice and Bob run the protocol with inputs x and y which might be correlated. In the key-agreement protocol Alice and Bob did not have local inputs for their next message generation algorithms. Eve, who is unaware of the exact values of x and y, needs to perform her attack oblivious of these values.

The second issue that arises is due to the nature of our attacks on undecomposable $\mathcal{F}$ as introduced in [MPR09]. These attacks crucially depend on the fact that the next message function of the parties is randomized function of the partial transcript generated so far and their local inputs. We shall call this property “Markov-chain property”, because the next message function of, say, Alice depends only on the partial transcript and her local input x and not on Bob’s view. But in presence of random oracles it is not guaranteed that the Markov-chain property holds. So, Eve needs to kill the dependencies between Alice and Bob views so that the Markov-chain property holds without knowing the explicit values of Alice and Bob’s local inputs.

Our eavesdropper algorithm works as follows. We assume that there are several input-less protocols $\pi(x, y)$, where Alice and Bob have local inputs fixed to x and y respectively. Our eavesdropper spawns one curious eavesdropper for *every* $(x, y) \in X \times Y$. If there is any query which is highly likely for some protocol $\pi(x, y)$ all eavesdroppers ask this query. We show that if there are no highly likely queries left in any of these protocols, then the Markov-chain property holds. Furthermore, such an eavesdropper is efficient. To show that this strategy is efficient, we need to prove stronger versions of the results included in [BM09].

Completeness vs. Security. Previously, we saw that it is possible to compile out the random oracle if parties have unbounded computational power (access to a PSPACE oracle suffices). For any protocol which is $(1 - \text{negl})$ complete and $(1 - \text{negl})$ semi-honest secure, we obtain a protocol which is $(1 - \varepsilon)$ complete and $(1 - \varepsilon)$ semi-honest secure. And the running time of the compiled protocol depends on $1/\varepsilon$. Although, ε could be driven down to arbitrary precision, it is still non-negligible. So, the compiled protocol has non-negligible insecurity and incompleteness. It is not evident whether a given undecomposable $\mathcal{F}$ could possible have such weakly semi-honest secure protocols. For example, coin tossing protocols could have arbitrary low insecurity if a sufficiently long protocol is used [Cle86].

The result in [MPR09] provides a gradual tradeoff between completeness and security. They show that as we increase completeness, the semi-honest insecurity of the protocol also increases. Thus, there is a unique ε^* such that $(1 - \varepsilon^*)$ complete protocol for $\mathcal{F}$ has $(1 - \varepsilon^*)$ semi-honest security, where $\varepsilon^* = 1/\text{poly}$ for some polynomial. Using this result, we can obtain a contradiction by using any $\varepsilon < \varepsilon^*$.

Generalized Oracles. The above mentioned results consider oracles which are restricted to random oracles. We can, in fact, generalize these oracles to much more general oracles, called atomic sub-modular oracles. These oracles can be used to capture several useful properties. For example, we can consider oracles which are injective length tripling random oracles. Moreover, we can also provide access to oracles which answer whether a particular point in the range of the random oracle has a pre-image or not. Showing that any atomic sub-modular oracle is useless for semi-honest secure computation is extremely useful in lifting our results to other oracles which show much more structure.

For example, consider the generic PKE-oracle with respect to which there exists a secure PKE protocol [GKM$^+$00].

1. Gen: Given a secret key sk, $\mathsf{Gen}(sk)$ is an injective length tripling random oracle which provides a public key pk.

2. Enc: Given a public key pk and a message m, $\mathsf{Enc}(pk, m)$ is an injective length tripling oracle which outputs the cipher text corresponding to encrypting m with the public key pk.

3. Test_1: Given a pk, it answers whether there exists sk such that $\mathsf{Gen}(sk) = pk$ or not.

4. Test_2: Given a pk and c, it answers whether there exits sk and m such that $\mathsf{Gen}(sk) = pk$ and $\mathsf{Enc}(pk, m) = c$.

5. Dec: This is the decryption oracle and dealing with this oracle needs slightly different approach (refer [MMP11]). Given sk and c, it outputs m such that $\mathsf{Gen}(sk) = pk$ and $\mathsf{Enc}(pk, m) = c$.

The quartet of oracles without the decryption oracle, $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Test}_1, \mathsf{Test}_2)$, shows significantly more structure than random oracles. We show that our compiler which shows that random oracles are useless for semi-honest secure function evaluation also works when we consider atomic sub-modular oracles. The result of [MMP11] which shows that PKE-oracle is useless for semi-honest secure function evaluation crucially relies on this result.

6.2 Random Oracles do not Help Semi-honest Secure Computation

In this section we will prove the following theorem:

Theorem 6. *Suppose there exists a* $(1 - \mathit{negl})$ *semi-honest secure protocol for an undecomposable* $\mathcal{F}$ *when parties have access to a random oracle and* PSPACE *oracle; then there exists a* $(1 - {}^1\!/_{\mathit{poly}})$ *semi-honest secure for* $\mathcal{F}$ *where parties have access to a* PSPACE *oracle and arbitrarily chosen polynomial poly.*

This theorem will be useful to show that random oracles are useless for semi-honest secure computation when deterministic two-party SSFE is considered. In other words, we will show that the assumption "There exists a semi-honest secure protocol for $\mathcal{F}$", where $\mathcal{F}$ is an undecomposable two-party deterministic SSFE, is black-box separated from OWF assumption. We will provide a proof which does not use the results in [BM09, DLMM11] in a black box manner. The main technique will involve transforming the secure protocol for $\mathcal{F}$ which uses $\mathcal{O}$ into one where parties have access to a PSPACE oracle but have no access to the random oracle. A result from [MPR09] shows that no undecomposable $\mathcal{F}$ can have a semi-honest secure protocol when parties have unbounded computational power (PSPACE oracle suffices).

Lemma 27 (Completeness vs. Security [MPR09])**.** *Let* $\mathcal{F}\colon X \times Y \mapsto \{0,1\}^*$ *be an undecomposable SSFE. Suppose* Π *is a two party protocol to evaluate* $\mathcal{F}$ *where Alice has an input* $x \in X$ *and Bob has an input* $y \in Y$*. If* Π *is a* $(1-\nu)$ *complete protocol then* Π *is* σ*-insecure, where*

$$\sigma \geq \left(1 - \frac{1}{2^{1/(|X|+|Y|)}}\right) \frac{(1 - 4|X||Y|\nu)}{4|X||Y|(|X|+|Y|)} \geq \frac{(1-4|X||Y|\nu)}{2(|X|+|Y|)^4}.$$

In particular, there exists a polynomial poly such that for any function $\mathcal{F}$ *which does not have a perfectly secure protocol, any protocol for evaluating* f *has a security error of at least* $1/\mathit{poly}(|X \times Y|)$*.*

We emphasize that this transformation is not round preserving. In this dissertation, we will present a proof which generalized results in [BM09] to show the result. The proof included in [MMP11], uses [BM09, DLMM11] reduces Theorem 6 to the results included in these prior works. We will assume that the reader is familiar with the proof techniques introduced in [BM09].

6.2.1 Notation

A *random oracle* is an infinite string $R = R_1 R_2 \ldots$, whose each bit is picked uniformly at random. When queried at a point q, the random oracle replies back with the bit R_q in $O(1)$ time. A $2n$ round two party protocol between Alice and Bob accessing a random oracle is said to be in *normal form* if it satisfies the following conditions:

1. During round $i \in [2n]$, Alice or Bob sends only one bit τ_i. Alice sends the first message and thereafter they send messages alternately, i.e. the bit τ_{2j-1} is sent by Alice τ_{2j} is sent by Bob.

2. Suppose i is odd and to generate the bit τ_i Alice does the following: Based on her local input x, local randomness r_A, transcript seen so far τ_1^{i-1} and her local query-answer list $I_A^{(i)}$, she deterministically generates the bit τ_i as instructed by the next message function of π. Similarly, Bob also generates the bit τ_i when i is even.

3. Alice and Bob never repeat a query, i.e. they always perform new queries in every round[1].

4. For any Alice and Bob input x and y, any arbitrary transcript is generated with non-zero probability[2].

If some two-party deterministic SSFE can be evaluated in a semi-honest securely using a random oracle, then we can assume that it also has a semi-honest secure protocol in the normal form.

Suppose Alice and Bob are running a $2n$ round normal form protocol π, i.e. Alice and Bob send n bits each. At the end of i rounds of the protocol π, the *view* of Alice comprises of: a) her input $x \in X$, b) local randomness r_A, c) the transcript τ_1^i, and d) her list of query answer-pairs represented by $I_A^{(i)}$. Similarly, Bob has input $y \in Y$, local randomness r_B and his list of query answer-pairs $I_B^{(i)}$. Then the view of Alice is $V_A^{(i)} = (x, \tau_1^i, r_A, I_A^{(i)})$ and Bob's view is $V_B^{(i)} = (y, \tau_1^i, r_B, I_B^{(i)})$.

We intend to define a consistency graph of views with respect to an eavesdropper Eve who has access to the random oracle and the transcript being generated by Alice and Bob. Let $I^{(i)}$ be the public set of query-answer pairs that an eavesdropper generated after seeing a transcript τ_1^i. Here Eve's view can be represented by the tuple $V_E^{(i)} = (\tau_1^i, I^{(i)})$. An execution of π is well defined by a tuple (x, y, r_A, r_B, R), where x, y are Alice and Bob inputs, r_A, r_B are their respective local randomness and R is the random oracle. We say that an Alice view $V_A^{(i)} = (x, \tau_1^i, r_A, I_A^{(i)})$ and a Bob view $V_B^{(i)} = (y, \tau_1^i, r_B, I_B^{(i)})$ are *consistent* with Eve's view $V_E = (\tau_1^i, I^{(i)})$ if, there exists a random oracle R such that the execution defined by the tuple (x, y, r_A, r_B, R) produces Alice, Bob and Eve views $(x, \tau_1^i, r_A, I_A^{(i)})$, $(y, \tau_1^i, r_B, I_B^{(i)})$ and $(\tau_1^i, I^{(i)})$ respectively.

An Alice view $V_A^{(i)}$ is consistent with Eve's view $V_E^{(i)}$, if there exists a Bob view $V_B^{(i)}$ such that $V_A^{(i)}$ and $V_B^{(i)}$ are consistent with $V_E^{(i)}$. Similarly, we can define consistency of a Bob view $V_B^{(i)}$ with Eve's view $V_E^{(i)}$. We define $\mathcal{A}_x^{(i)}$ as a multi-set of all Alice views $V_A^{(i)}$ which are consistent with the Eve's view $V_E^{(i)}$ and Alice has local input x, and the number of copies of view $V_A^{(i)}$ in $\mathcal{A}_x^{(i)}$ is proportional to $2^{-|I_A^{(i)} \setminus I^{(i)}|}$. In particular, we can consider a multi-set $\mathcal{A}_x$, where there are $2^{n-|I_A^{(i)} \setminus I^{(i)}|}$ copies of $V_A^{(i)}$. Intuitively, we want to weigh each $V_A^{(i)}$ proportional to the number of random oracles which are consistent with $V_A^{(i)}$ and $V_E^{(i)}$ which is

[1] If Alice was supposed to query at q and Alice had already queried q, then she queries the smallest query after q which she has not yet queried. This transformation should work for any protocol which permits curious parties.

[2] If the first n-bits in the local random tape of Alice is 0^n, then in the i-th round Alice queries the random oracle at i and sends the $(n+i)$-th bit in its local random tape. If the first half of the bits in the local random tape is not all 0s, then the random tape is used to run the original protocol. Similar modification can also be done for Bob.

$2^{N-|I_A^{(i)} \setminus I^{(i)}|}$.[3] Analogously, we define $\mathcal{B}_y^{(i)}$ as the multi-set of Bob views consistent with Eve's view where each view is suitably weighted.

A *consistency graph* with respect to Eve's view $V_E^{(i)}$, represented as $\mathcal{C}(V_E^{(i)})$, is a bipartite graph with partite sets $\mathcal{A}^{(i)} = \cup_{x \in X} \mathcal{A}_x^{(i)}$ and $\mathcal{B}^{(i)} = \cup_{y \in Y} \mathcal{B}_y^{(i)}$. We add an edge between $V_A^{(i)}$ and $V_B^{(i)}$ if $V_A^{(i)}$ and $V_B^{(i)}$ are consistent with respect to $V_E^{(i)}$. A projection of $\mathcal{C}(V_E^{(i)})$ to include only the vertices $\mathcal{A}_x^{(i)}$ and $\mathcal{B}_y^{(i)}$ is represented as $\mathcal{C}(V_E^{(i)})[\mathcal{A}_x^{(i)}, \mathcal{B}_y^{(i)}]$. One can make a simple observation that if $V_A^{(i)}$ is consistent with $V_E^{(i)}$ and $V_B^{(i)}$ is consistent with $V_E^{(i)}$ then $V_A^{(i)}$ and $V_B^{(i)}$ are consistent with respect to $V_E^{(i)}$ if and only if all queries in $I_A^{(i)} \setminus I^{(i)}$ and $I_B^{(i)} \setminus I^{(i)}$ are consistently answered.

A restricted notion of consistency was introduced in [BM09]. Two views $V_A^{(i)}$ and $V_B^{(i)}$ are *good* with respect to $V_E^{(i)}$ if $I_A^{(i)} \cap I_B^{(i)} \subseteq I^{(i)}$, i.e. all intersection queries in $I_A^{(i)}$ and $I_B^{(i)}$ are included in $I^{(i)}$. It is easy to see that goodness implies consistency, because if two views $V_A^{(i)}$ and $V_B^{(i)}$ are good with respect to i then they are also consistent with respect to $V_E^{(i)}$. Although it might be the case that there are consistent views such that the $I_A^{(i)} \cap I_B^{(i)} \not\subseteq I^{(i)}$. Now, we can define a *good-execution graph* with respect to Eve's view $V_E^{(i)}$, represented as $\mathcal{G}(V_E^{(i)})$. It is a subgraph of $\mathcal{C}(V_E^{(i)})$ where we add edges between two views $V_A^{(i)}$ and $V_B^{(i)}$ if they are good with respect to $V_E^{(i)}$. So, we exclude any edge in $\mathcal{C}(V_E^{(i)})$ which is between two views $V_A^{(i)}$ and $V_B^{(i)}$ consistent with $V_E^{(i)}$ but are not good. One important property of this representation is:

Lemma 28 ([BM09])**.** *The following two are identical distributions:*

1. *Picking a random edge from $\mathcal{G}(V_E^{(i)})[\mathcal{A}_x^{(i)}, \mathcal{B}_y^{(i)}]$ and return the corresponding Alice and Bob views.*

2. *Fix Alice and Bob inputs to be x and y respectively. Sample uniformly from the space of all executions such that after i rounds Alice and Bob views are good with respect to $V_E^{(i)}$ and the inputs of Alice and Bob are x and y, respectively. Return Alice and Bob views generated after running i rounds of the protocol π.*

This result can be used to uniformly sample from any good execution consistent with $V_E^{(i)}$. It is equivalent to uniformly picking an edge in $\mathcal{G}(V_E^{(i)})$ and returning the corresponding Alice and Bob views.

A (γ, δ)-*nice* Eve view $V_E^{(i)} = (\tau_1^i, I^{(i)})$ satisfies the following properties:

1. Low weight queries: For any $\mathcal{A}_x^{(i)}$, a query q occurs in at most γ fraction of the views in $\mathcal{A}_x^{(i)}$. Similarly, for any $\mathcal{B}_y$, a query q occurs in at most γ fraction of the views in $\mathcal{B}_y$.

[3] We are using $2^{n-|I_A^{(i)} \setminus I^{(i)}|}$ copies of $V_A^{(i)}$ instead of simply using $2^{N-|I_A^{(i)} \setminus I^{(i)}|}$ copies because in the first representation, there is a efficient representation for every vertex in $\mathcal{A}_x^{(i)}$ and in the latter case there is no efficient representation of the vertices in $\mathcal{A}_x^{(i)}$.

2. High connectivity: For any $\mathcal{A}_x^{(i)}$ and $\mathcal{B}_y^{(i)}$, a view $V_A^{(i)} \in \mathcal{A}_x^{(i)}$ is connected to at least $(1-\delta)$ fraction of the views in $\mathcal{B}_y^{(i)}$. Similarly, for any $\mathcal{A}_x^{(i)}$ and $\mathcal{B}_y^{(i)}$, a view $V_B^{(i)} \in \mathcal{B}_y^{(i)}$ is connected to at least $(1-\delta)$ fraction of the views in $\mathcal{A}_x^{(i)}$.

Intuitively, it means that there is no high probability query in either Alice or Bob view in $\mathcal{G}(V_E^{(i)})$ and the joint-distribution of Alice-Bob views is close to a product distribution.

6.2.2 Argument

Suppose we have executed i rounds of a normal form protocol π and Alice is supposed to send the next bit in the protocol. In this section, we assume that Eve has generated a (γ, δ)-nice view $V_E^{(i)} = (I^{(i)}, \tau_1^i)$ and in the next section we will see how to generate such an Eve view inductively. For any view $V_A^{(i)}$ in $\mathcal{A}^{(i)}$, we can deterministically compute the next query q that Alice will perform. If this query lies in $I_B^{(i)} \setminus I^{(i)}$, when $V_B^{(i)}$ is uniformly chosen from $\mathcal{B}_y^{(i)}$, then we say that event $\texttt{Inter-turn-Fail}_i$ has occurred. Conditioned on this event not occurring, there are two cases to consider:

1. **If q is already included in $I_A^{(i)} \cup I^{(i)}$:** Since the protocol is in the normal form, $q \in I^{(i)} \setminus I_A^{(i)}$. Let $I_A^{(i)}$ be the local query-answer list of Alice and $I_A^{(+)}$ be the local query-answer list *after* the query-answer pair (q, b) is added to $I_A^{(i)}$, where $b = R_q$ and $(q, b) \in I^{(i)}$. We denote the new Alice view as $V_A^{(+)}$. Observe that $V_A^{(+)}$ and $V_B^{(i)}$ are good with respect to $V_E^{(i)}$ if and only if $V_A^{(i)}$ and $V_B^{(i)}$ are good with respect to $V_E^{(i)}$. So, $V_A^{(+)}$ is connected to at least $(1-\delta)$ fraction of any $\mathcal{B}_y^{(i)}$, for $y \in Y$.

2. **If q is not included in $I_A^{(i)} \cup I^{(i)}$:** In this case, from the current public query-answer list generated by Eve and from Alice's view, querying the random oracle at q is equally likely to return 0 and 1. Observe that if $V_A^{(i)}$ and $V_B^{(i)}$ are good with respect to $V_E^{(i)}$ and q is not queried in $V_B^{(i)}$, i.e. $q \notin I_B^{(i)}$, then $V_A^{(+)}$, regardless of the random oracle's answer b, and $V_B^{(i)}$ are also good with respect to $V_E^{(i)}$. The view $V_A^{(i)}$ and at least $(1-\gamma-\delta)$ fraction of Bob views in $\mathcal{B}_y^{(i)}$ are good with respect to $V_E^{(i)}$ and $q \notin I_B^{(i)}$, because q occurs in less than γ fraction of views of $\mathcal{B}_y^{(i)}$.

Now, for every Alice view $V_A^{(+)}$, we can deterministically ascertain whether the next bit τ_{i+1} in the protocol is 0 or 1.

We are interested in computing an intermediate graph to capture the view of Alice and Bob executions just prior to sending τ_{i+1} which are good with respect to $V_E^{(i)}$. If $V_A^{(i)}$ was such that $q \in I^{(i)}$ then for every copy of $V_A^{(i)} \in \mathcal{A}^{(i)}$ we make a copy of $V_A^{(+)}$ (this is because $I_A^{(+)} \setminus I_E^{(i)}$ and $I_A^{(i)} \setminus I_E^{(i)}$ are identical). Otherwise, for every two copies of $V_A^{(i)}$ we make one copy of $V_A^{(+)}$ with $b = 0$ and another with $b = 1$ (this is because $|I_A^{(+)} \setminus I_E^{(i)}| - |I_A^{(i)} \setminus I_E^{(i)}| = 1$). Similar to the notation earlier, let this set of Alice views be called $\mathcal{A}^{(+)}$ and those with Alice input x be called $\mathcal{A}_x^{(+)}$. Basically, it is the multi-set of all Alice views just prior to sending τ_{i+1} with appropriate multiplicity for every view. We consider the bipartite graph $\mathcal{G}^{+}(V_E^{(i)})$ with partite sets $\mathcal{A}^{(+)}$ and $\mathcal{B}^{(i)}$ with edges between $V_A^{(+)}$ and $V_B^{(i)}$ if they are good with respect to $V_E^{(i)}$. Let $\alpha_{I^{(i)}}(x, \tau_1^i)$ be the fraction of Alice views in $\mathcal{A}_x^{(+)}$ which sends 0 as the next bit in the protocol. For any arbitrary $x \in X$ and $y \in Y$, it can easily be concluded that:

$$(1-\gamma-\delta) \leq \frac{\Pr_{(V_A,V_B)\leftarrow\mathcal{G}^{(+)}(V_E^{(i)})[\mathcal{A}_x^{(+)},\mathcal{B}_y^{(i)}]}[\tau_{i+1}=0]}{\alpha_{I^{(i)}}(x,\tau_1^i)} \leq \frac{1}{(1-\gamma-\delta)}$$

We use the notation $u \approx_\mu v$ to represent $\min\{\mu, {}^1/_\mu\} \leq {}^u/_v \leq \max\{\mu, {}^1/_\mu\}$. So, we have shown that

$$\Pr_{(V_A,V_B)\leftarrow\mathcal{G}^{(+)}(V_E^{(i)})[\mathcal{A}_x^{(+)},\mathcal{B}_y^{(i)}]}[\tau_{i+1}=0] \approx_{(1-\gamma-\delta)} \alpha_{I^{(i)}}(x,\tau_1^i)$$

Observe that the quantity $\alpha_{I^{(i)}}(x, \tau_1^i)$, for every $x \in X$, can be estimated by Eve if she has unbounded computational power. So, we can conclude the following:

Lemma 29. *Conditioned on the event that the execution has remained "good" for every round $r \in [i]$*

1. *The probability of the event* `Inter-turn-Fail`$_i$ *is at most γ.*

2. *Conditioned on the event that* `Inter-turn-Fail`$_i$ *does not happen, the probability of Alice sending next bit $\tau_{i+1} = 0$ is within a multiplicative factor of $(1-\gamma-\delta)$ from $\alpha_{I^{(i)}}(x, \tau_1^i)$.*

6.2.3 Eve Strategy

We follow the notation introduced in [BM09] to explain our Eve strategy, represented by Eve*. With respect to an Eve view $V_E^{(i)}$, a query q is *heavy for an input pair* (x, y) if $\Pr_{(V_A^{(i)},V_B^{(i)})\leftarrow\mathcal{G}(V_E^{(i)})[\mathcal{A}_x^{(i)},\mathcal{B}_y^{(i)}]}[q \in I_A^{(i)} \cup I_B^{(i)}] \geq {}^\varepsilon/_n$.

1. Initialize list $\mathcal{L}$ to an empty set. The list $\mathcal{L}$ will contain all inputs (x, y) such that, over the history of generation of $V_E^{(i)}$, Eve tried to perform the $(1 + {}^{n^2}/_{\varepsilon^2})$-th heavy query in $\mathcal{G}(V_E^{(i)})[\mathcal{A}_x^{(i)}, \mathcal{B}_y^{(i)}]$.

2. Choose q which is heavy for some (x, y) such that $(x, y) \notin \mathcal{L}$. Increment the count of heavy queries for every $(x', y') \notin \mathcal{L}$ if q is heavy for (x', y'). For every (x', y'), if q is $(1 + {}^{n^2}/_{\varepsilon^2})$-th heavy query for (x', y'), then add (x', y') to $\mathcal{L}$. If the set $X \times Y \setminus \mathcal{L}$ is non-empty, then query the random oracle at q; otherwise abort.

We can import the following result form [BM09]:

Lemma 30 (Highly like to be Good [BM09])**.** *Conditioned on the fact that Eve* does not abort, an execution will remain "good" for every round of π with probability at least* $(1-3\varepsilon)$.

At the end of a round, if we have no heavy queries for any input pair $(x, y) \in X \times Y$, then it implies a $(2\varepsilon/n, 2\varepsilon)$-nice Eve view. By definition, the Eve* algorithm presented here is efficient[4]. It aborts if it tries to perform $|X||Y|(1 + {n^2}/{\varepsilon^2})$ queries. So, all we need to ensure is that the probability of Eve* aborting is small.

In the remainder of the section, we will introduce the notion of eavesdroppers who are allowed to perform only bounded number of heavy queries. Then we will show that we can come up with an algorithm Eve$_\pi$ which uses these eavesdroppers to mimic the behavior of Eve* and it fails with probability at most 4ε.

Alternate Eve Description. We introduce a particular class of Eve strategies, called *bounded-query* Eve strategy. Let ρ be an input less protocol in normal form, i.e. Alice and Bob locally honestly pick random tapes r_A and r_B and they are allowed to access the random oracle once every round. An Eve algorithm in the class Bounded-Eve(B, ρ) satisfies the following conditions:

1. The algorithm of Eve can follow any arbitrary mechanism to generate the queries for the public query-answer list $I^{(i)}$.

2. At any particular round $i \in [2n]$, we consider the good executions of ρ consistent with Eve view $V_E^{(i)}$. A query q is classified as "heavy" if, with respect to the current Eve view $V_E^{(i)}$, q is a query which occurs with probability at least ε/n. The only restriction is that, at the end of each round i when Eve stops querying there should be no more heavy queries left in the good-execution graph of ρ.

3. The algorithm is permitted to perform at most B heavy queries. If it tries to perform $(B+1)$-th heavy queries then it returns `Failure`.

4. Eve is terminated if all queries consistent with the current Eve view, at the end of round $2n$ of the protocol ρ, are not heavy or if it returns `Failure`. If Eve terminates without failure, it returns `Success`.

Basically, in this class we are trying to capture Eve algorithms which might perform several light queries in between the heavy queries; but we charge the algorithm only when heavy queries are performed. The result in [BM09] can be generalized to the following:

[4] Explicitly, Eve* performs a query q if:

1. The query q is heavy for some input pair $(x, y) \notin \mathcal{L}$, and
2. There exists an input pair $(x', y') \notin \mathcal{L}$ such that: Either q is not heavy with respect to (x', y'), or q is at most the ${n^2}/{\varepsilon^2}$-th heavy query for (x', y') over the history of generation of $V_E^{(i)}$.

Lemma 31 (Heavy Queries are never Useless [BM09]). *For any two party protocol ρ, when ρ is honestly run, any Bounded-Eve$({}^{n^2}/_{\varepsilon^2}, \rho)$ outputs* `Failure` *with probability at most 4ε.*

Proof. The bound on the event "Fail" still remains identical.

For the remainder of the proof, we can define the variables suitably and keep the analysis identical as in [BM09]. For example, we need to make the following changes:

1. Long is the event that Eve makes more than ${}^{n^2}/_{\varepsilon^2}$ heavy queries.

2. Y_j as indicator variable for the event that Eve sends at least j heavy queries.

3. p_j as the probability that Eve asks the j-th heavy query.

4. Y_j^q as the indicator variable for the j-th heavy query asked by Eve is q and q was asked before by Alice or Bob.

The whole proof remains identical. □

Here we are considering *arbitrary* Eve strategies; but the guarantee of "Fail" probability being small is dependent on the fact that transcripts were generated based on honest execution of ρ. Intuitively, the lemma says that there is no way an Eve can make more than ${}^{n^2}/_{\varepsilon^2}$ heavy queries and the event that the good-execution graph still has a heavy query remaining happens with probability more than 4ε.

Our construction of Eve_π for the protocol π honestly running with *any* input $x^* \in X$ and $y^* \in Y$ is as follows[5]:

1. It will spawn an algorithm $\text{Eve}_{(x,y)}$, for every Alice input x and Bob input y, in the class Bounded-Eve$({}^{n^2}/_{\varepsilon^2}, \pi(x,y))$. Here $\pi(x,y)$ is the input-less protocol π where Alice and Bob have their inputs fixed to x and y respectively, and both parties uniformly choose their local randomness and follow the protocol π accessing a random oracle R.

2. List $\mathcal{L}$ is initialized to an empty set. List $\mathcal{L}$ is going to store inputs pairs (x,y) such that $\text{Eve}_{(x,y)}$ were terminated when it tried to perform its $(1 + {}^{n^2}/_{\varepsilon^2})$-th heavy query. We will assume that if Eve_π requests an $\text{Eve}_{(x,y)}$, for some $(x,y) \in \mathcal{L}$, to query the random oracle it keeps returning `Failure`. This list will be updated every time Eve_π decides to query the random oracle. For account keeping purposes, we will additionally define $\mathsf{last}(x,y)$ as the $(1 + {}^{n^2}/_{\varepsilon^2})$-th heavy query that $\text{Eve}_{(x,y)}$ tried to perform.

[5] The construction of Eve_π is oblivious to the exact x^* and y^* values.

3. If there exists an input pair $(x', y') \notin \mathcal{L}$ and a query q such that q is heavy for input pair (x', y') then Eve_π requests every $\text{Eve}_{(x,y)}$ to query q. If some $\text{Eve}_{(x,y)}$ queries the random oracle at q, i.e. not all $\text{Eve}_{(x,y)}$s output `Failure`[6], then we add the query answer pair (q, R_q) to the public query list $I^{(i)}$. We add entries to the list $\mathcal{L}$ to include any (x, y) such that $\text{Eve}_{(x,y)}$ returned `Failure` for the first time and suitably add $\mathsf{last}(x, y)$ entries for them. If every $\text{Eve}_{(x,y)}$ outputs `Failure` then Eve aborts the protocol and outputs `Failure_overall`. Every round, we repeat these steps till there are no heavy queries for any input pair (x, y) left in the graph $\mathcal{G}(V_E^{(i)})[\mathcal{A}_x^{(i)}, \mathcal{B}_y^{(i)}]$. And if we are able to successfully complete this algorithm for every round $i \in [2n]$ of the protocol π then return `Success_overall`.

Bounding the failure probability: It is easy to see that Eve_π queries at most $|X||Y|(1 + n^2/\varepsilon^2)$ queries, because every query is heavy for some input not in $\mathcal{L}$ and we never delete an entry from $\mathcal{L}$.

Consider an execution of Eve_π where it outputs `Failure_overall`, then we know that every entry $X \times Y$ is in $\mathcal{L}$. In particular (x^*, y^*) is in $\mathcal{L}$. This implies that just prior to performing $\mathsf{last}(x^*, y^*)$ there was a heavy query left in $\mathcal{G}(V_E^{(i)})[\mathcal{A}_{x^*}^{(i)}, \mathcal{B}_{y^*}^{(i)}]$. Consider Eve' which runs Eve_π and terminates just before performing the query $\mathsf{last}(x, y)$. Eve' is in the class Bounded-Eve$(n^2/\varepsilon^2, \pi(x^*, y^*))$. This shows that `Failure_overall` is at most 4ε; otherwise we will get a contradiction.

Finally, it is trivial to see that Eve^* aborts if and only if Eve_π outputs `Failure_overall`; hence we conclude that Eve^* aborts with probability at most 4ε.

6.2.4 Putting Everything Together

Consider the following π^* algorithm: We run Eve^* after every round of the protocol. To build the protocol tree $\mathcal{T}_{(x,y)}$, for every partial transcript τ_1^i, we define the probability of next bit being 0 to be $\alpha_{I^{(i)}}(x, \tau_1^i)$ if it is an Alice node. Similarly, for a Bob node, we define the probability of next bit being 0 as $\beta_{I^{(i)}}(y, \tau_1^i)$. For every edge in the protocol tree we have a probability associated to make that transition and every node has an associated query-answer list generated by Eve. In these protocol trees, the random oracle is only accessed by Eve^*. Alternately, Eve^* can simulate such an random oracle in her head, i.e. when she needs to query the random oracle at q, she uniformly samples a random oracle R consistent with her current view and pretends that $R(q)$ is the outcome of the random oracle when queried at q. Moreover, Eve^*'s strategy is deterministic, so Alice or Bob could implement the Eve^* strategy, thus providing a protocol $\mathcal{T}_{(x,y)}$ in the plain model.

Using simple union bound, we can claim that the distribution of transcripts generated in the simulation of π by π^* and the distribution of transcripts as generated by π have statistical distance at most $\Theta(\varepsilon)$.

[6] Every $\text{Eve}_{(x,y)}$ such that $(x, y) \in \mathcal{L}$ always returns `Failure`.

Hence, we obtain Theorem 6. If the original protocol, where parties have access to the random oracle and the PSPACE oracle, was $(1-\mathrm{negl}(n))$ complete and $(1-\mathrm{negl}(n))$ semi-honest secure, then the final protocol, where parties have access to a PSPACE oracle, is $(1-{}^{1}\!/_{\mathrm{poly}(n)})$ complete and $(1-{}^{1}\!/_{\mathrm{poly}(n)})$ semi-honest secure for arbitrary choice of polynomial poly.

6.3 Black-Box Separations

In this section we prove the blackbox separations implied by our impossibility result in the random oracle model.

Theorem 7. *Suppose $\mathcal{F}\colon X \times Y \mapsto Z$ is a function which is not decomposable (and thus does not have a perfectly secure two-party protocol) and $|X| \cdot |Y| = poly(n)$ where n is the security parameter. Then there is no secure black-box two-party protocol for computing $\mathcal{F}$ based on the primitive P if P is one of the following: one-way function, one-way permutation, collision resistant hash function, block-cipher (including exponentially hard versions of these primitives). In fact, P can be any primitive that can be constructed from random oracle (or ideal cipher) in a black-box manner.*

In this section we will assume that the random oracle is a random function from $\{0,1\}^n$ to $\{0,1\}^n$, where n is the security parameter. Our previous results, where we interpreted a random oracle as a long string with every bit independently and uniformly generated, also extend to this model of random oracles.

The Case of $P =$ one-way functions (OWF). Let $\mathcal{F}$ be the function described in the statement of the theorem. Suppose $C^{\mathcal{O}}$ is a black-box construction of a semi-honest secure evaluation of $\mathcal{F}$ with completeness $1 - \mathrm{negl}(n)$, given black-box access to any one-way function $\mathcal{O}$. Let S be the efficient black-box security reduction that for every oracle $\mathcal{O}$, and every adversary A against $C^{\mathcal{O}}$ which violates the semi-honest security with non-negligible probability, $S^{\mathcal{O},A}$ inverts $\mathcal{O}$ with non-negligible probability. We feed the construction a random oracle $\mathcal{O}$ as input. Since, $\mathcal{F}$ is undecomposable we know from Theorem 6 and Lemma 27 that there exists an adversary A' which violates the semi-honest security of the construction with non-negligible probability ε. Formally, over all choices of random oracles and the random tapes of this adversary A' the probability that it violates the security the protocol is at least ε. This implies that there are ${}^{\varepsilon}\!/_{2}$ fraction of random oracles such that A' violates the security of the protocol with probability at least ${}^{\varepsilon}\!/_{2}$. For each such random oracle, we can use the reduction $S^{\mathcal{O},A'}$ to invert the one-way function with non-negligible probability. Thus, we have an efficient algorithm which performs only $\mathrm{poly}(n)$ queries to the random oracle and inverts it with non-negliible probability. But this is a contradiction, because any, possibly computationally unbounded, algorithm who asks at most $k = 2^{o(n)}$ queries to a random oracle, is able to invert a random image $\mathcal{O}(\mathbf{U}_n)$ only with probability $\leq k/2^n = 2^{-\Omega(n)} = \text{negl}(n)$.

The Case of $P =$ collision resistant hashing (CRH). The case for collision resistant hash functions is very similar to that of one-way function. In fact, the same argument as for the case of one-way function can be used for any primitive that can be derived in an information theoretically secure way, from random oracle model. An implementation of a CRH h from a random oracle $\mathcal{O}$ is very easy: $h(x) = \mathcal{O}(x)_1^{\lceil |x|/2 \rceil}$, where y_i^j represents the i to j bits of the string y. Then, again, the black-box implementation reduction allows us to feed the construction with a random oracle, and the black-box security reduction gives a way to break the random hash function with only poly(n) queries, which is impossible.

The Case of $P =$ one-way permutation (OWP). Here, the argument relies on a fact, first employed by Impagliazzo and Rudich [IR89]. If the honest parties of a protocol together with an adversary attacking this protocol protocols in the random oracle model ask only k oracle queries, then the very same attacker will succeed also in the random *permutation* oracle as well, as long as all the parties ask their oracle queries only from domains of size at least $\gg k^2$. More formally, asking k queries from a random orale over a domain of size αk^2 will lead to a collision with probability at most $k^2/(\alpha k^2) = {}^1\!/\!_\alpha$, and thus this experiment can not differentiate the random oracle from a random permutation with an advantage more than ${}^1\!/\!_\alpha$. If the original attack in the random oracle model succeeds with advantage $\varepsilon = 1/\text{poly}(n)$, then the adversary in the random permutation oracle could use $\alpha = {}^{10}\!/\!_\varepsilon$ and ask all of the permutation queries for domains of size $1, 2, \ldots, \alpha \cdot k^2$ (which is a total of only $1 + 2 + \cdots + \alpha k^2 = O(\alpha^2 k^4) = \text{poly}(n)$ many queries). This way, the adversary can pretend that the parties are asking their queries only from the domains of size at least αk^2 (and smaller domains are public knowledge). This modified attack will break the scheme in the random permutation oracle with advantage at least $\varepsilon - {}^\varepsilon\!/\!_{10} > {}^\varepsilon\!/\!_2$ which is still noticeable. The rest of the argument is similar to the case of one-way function. Namely, to finish the proof we observe that, similar to the case of a random function, any poly(n)-query algorithm is able to invert a random *permutation* $\mathcal{O}$ only with a negligible chance.

The Case of $P =$ block cipher. Note that an ideal-cipher oracle can be used as a black-box to get a block-cipher with exponential security, against adversaries who ask only poly(n) oracle queries. Therefore, similar to the cases above, we would be done if we could break the security of any candidate black-box protocol for computing $\mathcal{F}$ in the ideal-cipher model, with poly(n) number of oracle queries. It would suffice if we could show how to to "simulate" an ideal cipher oracle IC using access to a random oracle $\mathcal{O}$ (which, for example, was trivial for the case of CRH). Simulating an ideal cipher oracle using a random oracle, however, is a highly nontrivial task, and it was only a few years ago [CPS08] that finally such a construction was proved.[7] More formally, the result of [CPS08, HKT11] shows that for any construction C_{IC} of some cryptographic primitive in the ideal cipher model, there is a closely related construction in the random oracle model $C^{\mathcal{O}}$, such that if the latter construction $C^{\mathcal{O}}$ can be broken by an adversary $A^{\mathcal{O}}$ with k queries and ε advantage, then the former construction C^{IC} can be also broken by another adversary A^{IC} who asks only $\mathrm{poly}(k, n, 1/\varepsilon)$ oracle queries (to the ideal cipher oracle) and breaks the scheme with advantage $\mathrm{poly}(1/k, 1/n, \varepsilon)$. Therefore, by taking the adversary $A^{\mathcal{O}}$ to be our attacker of Theorem 7, we can derive an query-efficient attacker A' against any protocol, based on the ideal-cipher oracle. As we said before, this is sufficient for deriving the black-box separation against block ciphers (as well as any primitive which is implied by an ideal-cipher in an information theoretic way).

6.4 Generalized Oracles

In this section we generalize our result for random oracles to a larger class of oracles. As before, an oracle $\mathcal{O}$ is a function from inputs to outputs, which is chosen according to a distribution at the beginning of the execution and kept fixed. However, unlike in the case of random-oracles, now we will allow a distribution such that the output at a certain input can depend on outputs at other inputs, in a limited way. A set of query-answer pairs X is called *valid* if there is a positive probability that for every $(q, a) \in X$, $\mathcal{O}(q) = a$. The probability $\Pr[\mathcal{O}(q) = a \mid X]$ represents the probability of $\mathcal{O}(q)$ being answered as a when $\mathcal{O}$ is sampled, conditioned on being consistent with the (valid) set of query-answer pairs X. We define the *closure* of a set of query-answer pairs X, represented by X^*, as the largest set of all query-answer pairs which get *fixed* when we condition that $\mathcal{O}$ is consistent with X. Formally, $X^* = \{(q, a) \mid \Pr[\mathcal{O}(q) = a \mid X] = 1\}$. (Considering functions as sets of query-answer pairs, X^* is the intersection of all functions in the support of $\mathcal{O}$ that are consistent with the query-answer pairs in X.) A query q is *answered* in a set of query-answer pairs X, if the entry (q, a) exists in the set X, for some a.

Definition 2 (Atomic Sub-modular Oracles)**.** *$\mathcal{O}$ is a* atomic sub-modular oracle *if, for any two query-answer sets X, Y such that $X \cup Y$ is valid and $|X^*|, |Y^*|$ are at most poly(n), any query q not answered in $X^* \cup Y^*$ and any answer a in the range of the functions in the support of $\mathcal{O}$,*

$$\Pr[\mathcal{O}(q) = a \mid X \cup Y] \approx_{(1+\nu)} \Pr[\mathcal{O}(q) = a \mid X].$$

[7] The presented construction was simply a six-round Fiestel network with a very complicated proof of security. However, Holenstein et al. [HKT11] showed that the proof of [CPS08] is incorrect. But, they showed that using fourteen rounds instead of six-round Fiestel network would remedy the proof.

We will provide a slightly general definition later but the results proven in this section will carry over to the more general definition as well.

That is, if $\mathcal{O}$ is a atomic sub-modular oracle, and a query is answered neither in X^* nor in Y^*, then $X \cup Y$ contains virtually no information about the query. Here sub-modularity refers to the fact that two sets X and Y cannot be combined to obtain information that is not already given by X or Y, and atomicity refers to the fact that (by considering $X = \varnothing$ above) a query-answer set Y either fixes the answer for q or reveals virtually no information about q.

It is trivial to see that random oracles are atomic sub-modular oracles with $\nu = 0$. But there are other useful examples too. To show the result that a generic PKE oracle cannot help perform semi-honest secure computation of undecomposable SSFE, we will depend on the following oracle (represented by a quartet of oracles $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Test}_1, \mathsf{Test}_2)$) being a atomic sub-modular oracle:

- Gen: It is a length-tripling random oracle from the set of inputs $\{0,1\}^n$ to $\{0,1\}^{3n}$. It takes as input a secret key sk and provides a public-key pk corresponding to it, i.e. $\mathsf{Gen}(sk) = pk$.

- Enc: It is a length-tripling random oracle from the set of inputs $\{0,1\}^{4n}$ to $\{0,1\}^{12n}$. It takes as input a (possibly invalid) public key pk and a message m and provides the corresponding cipher text c 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 output 1 if and only if there exists sk and m such that $\mathsf{Gen}(sk) = pk$ and $\mathsf{Enc}(pk, m) = c$.

We note that the encryption oracle produces cipher texts for public keys pk irrespective of whether there exists sk satisfying $\mathsf{Gen}(sk) = pk$. If we additionally provide access to a decryption oracle Dec such that $\mathsf{Dec}(sk, c) = m$, where $\mathsf{Gen}(sk) = pk$ and $\mathsf{Enc}(pk, m) = c$, then we can perform public key encryption using this oracle. In [MMP11] we show that even this oracle is useless for semi-honest deterministic SSFE. But, that result crucially depends of the fact that the quartet of oracles mentioned above are useless for semi-honest deterministic SSFE. We next show that this quartet of oracles is a atomic sub-modular oracle.

6.4.1 Some Examples

Let us consider some other non-trivial examples of atomic sub-modular oracles. Consider the following pair of oracles $(\mathcal{R}, \mathsf{Test})$, where

1. $\mathcal{R}$ is a random oracle from domain $\{0,1\}^n$ to range $\{0,1\}^{3n}$, and

2. Test is a boolean function from $\{0,1\}^{3n}$ to $\{0,1\}$; and $\mathsf{Test}(a) = 1$ if and only if there exists q such that $\mathcal{R}(q) = a$.

The closure of sets for this oracle is defined in the following manner:

1. $\{(\mathcal{R}(q), a)\}^* = \{\ (\mathcal{R}(q), a),\ (\mathsf{Test}(a), 1)\ \}$,

2. $\{(\mathsf{Test}(a), b)\}^* = \{\ (\mathsf{Test}(a), b)\ \}$ where $b \in \{0,1\}$, and

3. If X is valid, then $X^* = \cup_{x \in X}\ \{x\}^*$.

We point out that if an oracle $\mathcal{O}$ satisfies the property that $(X \cup Y)^* = X^* \cup Y^*$, and for any q not answered in X^*, $\Pr[\mathcal{O}(q) = a \mid X] \approx_{(1+\nu)} \Pr[\mathcal{O}(q) = a]$, then $\mathcal{O}$ is an atomic sub-modular oracle. Since the oracle $(\mathcal{R}, \mathsf{Test})$ does indeed satisfy the first property, we turn to proving the second one:

Claim 2. *Consider a valid set of query-answer pairs X. For any $a \in \{0,1\}^{3n}$ not answered in X^*, we have $\Pr[\mathit{Test}(a) = 1 \mid X^*] \approx_{(1+\nu)} \Pr[\mathit{Test}(a) = 1]$.*

Proof. Without loss of generality, we can assume that X consists only of $\mathcal{R}$ queries and query-answer pairs of the form $\mathsf{Test}(y) = 0$.[8] Suppose there are δ queries to the $\mathcal{R}$ oracle and λ queries to the Test oracle in the set X. Conditioned on random oracles which are consistent with the query-answer pairs in X, the probability of $\mathsf{Test}(a) = 1$ is:

$$1 - \left(1 - \frac{1}{2^{3n} - \lambda - 1}\right)^{2^n - \delta}$$

Define the function:

$$f_N(x, y) = 1 - \left(1 - \frac{1}{N^3 - y}\right)^{N - x}$$

To prove the lemma, it suffices to show a negligible upper bound, for $\delta, \lambda = \mathrm{poly}(n)$, on the quantity

$$\frac{(f_{2^n}(0,0) - f_{2^n}(\delta, \lambda))}{f_{2^n}(0,0)}$$

Using Lemma 47, we get an upper bound of

$$\frac{\delta + o(1)}{2^n - o(1)} \leq \frac{1}{2^{n/2}} = \nu$$

Using this value of ν we can satisfy the definition of atomic sub-modular oracle. □

[8] If there are $\mathsf{Test}(y) = 1$ queries in the set X we can replace each of them with a new $\mathcal{R}(x_y) = y$ query.

Observe that the probability of a query-answer pair $(\mathsf{Test}(a), 0)$, where a is not answered in P, is close to 1. The above mentioned result for the query-answer pairs of the form $(\mathsf{Test}(a), 1)$ is sufficient to derive an analogous result for $(\mathsf{Test}(a), 0)$ query-answer pairs.

Using the same argument, we can show that in the oracle $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Test}_1, \mathsf{Test}_2)$ Test_1 queries satisfy the atomic sub-modular oracle definiton. Also, we can use the same technique, along with a union bound on $m \in \{0,1\}^n$, to show that Test_2 queries also satisfy the atomic sub-modular oracle definition.

6.4.2 Generalized Independence Learners

In this section, we will show that an approximate version of the product characterization of protocols using random oracles in [BM09] can be extended to protocols with access to atomic sub-modular oracles. Consider a two party oracle protocol where parties have access to an oracle $\mathcal{O}$. Suppose the eavesdropper Eve has produced a public query-answer list I and the transcript generated so far is M. We say that an execution is *Good* with respect to (M, I) if all random oracles queries in the intersection of Alice and Bob views, I_A and I_B respectively, lie in the Eve query-answer list I, i.e. $I_A \cap I_B \subseteq I$. We can generalize this notion of good executions to atomic sub-modular oracles and an execution is called good with respect to (M, I^*) if intersection of ${I_A}^*$ and ${I_B}^*$ is contained in I^*. The distribution $\mathcal{GEXEC}(M, I^*)$ is the distribution of good executions where Eve's query-answer list is I^* and the transcript generated by the protocol is M. When $\mathcal{O}$ is a random oracle, it was shown in [BM09] that $\mathcal{GEXEC}(M, I^*) = (\mathcal{A}_{I^*} \times \mathcal{B}_{I^*}) \mid \mathsf{Good}(M, I^*)$, where $\mathsf{Good}(M, I^*)$ is true if the execution is good with respect to the transcript M and Eve query-answer list I^*. Instead of the "equality" in this equation, we can claim an approximate version of this statement for atomic sub-modular oracles. We can show that the two terms are within a $(1 + \mathrm{negl}(n))$ multiplicative factor of each other.

Lemma 32 (Generalization of Product Characterization in [BM09])**.** *Let $\mathcal{O}$ be an atomic sub-modular oracle. There exists a negligible function $\tilde{\nu}$, such that:*

$$\mathcal{GEXEC}(M, I^*) \approx_{(1+\tilde{\nu})} (\mathcal{A}_{I^*} \times \mathcal{B}_{I^*}) \mid \mathsf{Good}(M, I^*)$$

Proof. Consider a pair of Alice and Bob views (A, B) which are good with respect to (M, I^*). So, if the queries in A and B are represented by I_A and I_B respectively, then the probability of sampling (A, B) according to the distribution $\mathcal{GEXEC}(M, I^*)$ is proportional to:

$$\Pr[{I_A}^*, {I_B}^* \mid I^*] \;=\; \Pr[{I_A}^* \mid I^*] \times \Pr[{I_B}^* \mid I^*, {I_A}^*]$$

The only nontrivial step in this theorem is to use the property of atomic sub-modular oracles to show that $\Pr[{I_B}^* \mid I^*, {I_A}^*] \approx_{(1+\tilde{\nu})} \Pr[{I_B}^* \mid I^*]$. Let $Y = {I_B}^* = \{e_1, \ldots, e_k\}$ and $Y^{(i)} = \{e_1, \ldots, e_{i-1}\}$. We will show that $\Pr[e_i \mid Y^{(i)}, I^*, {I_A}^*] \approx_{(1+\nu)} \Pr[e_i \mid Y^{(i)}, I^*]$.

1. If $e_i \in {(Y^{(i)} \cup I^*)}^*$: In this case $\Pr[e_i \mid Y^{(i)}, I^*, {I_A}^*] = \Pr[e_i \mid Y^{(i)}, I^*] = 1$.

2. If $e_i \in {I_A}^* \setminus {(Y^{(i)} \cup I^*)}^*$: This case is not possible, because we know that ${I_A}^* \cap {I_B}^* \subseteq I^*$.

3. If $e_i \notin {I_A}^* \cup {(Y^{(i)} \cup I^*)}^*$: In this case we can apply the property of atomic sub-modular oracle to obtain:
$$\Pr[e_i \mid {I_A}^* \cup {(Y^{(i)} \cup I^*)}^*] \approx_{(1+\nu)} \Pr[e_i \mid {(Y^{(i)} \cup I^*)}^*] = \Pr[e_i \mid Y^{(i)}, I^*]$$

This implies that $\Pr[{I_B}^* \mid I^*, {I_A}^*] \approx_{(1+\nu)^{|{I_B}^*|}} \Pr[{I_B}^* \mid I^*]$ and $\tilde{\nu} = (|{I_B}^*| + 1)\nu$ suffices, since $|{I_B}^*|$ is at most poly(n).

The distribution $\mathcal{A}_{I^*}$ is defined as follows: For any Alice view A, the probability of sampling A is proportional to $\Pr[{I_A}^* \mid I^*]$, where I_A is the set of query-answer pairs in the Alice view A. Similarly, we can define the distribution $\mathcal{B}_{I^*}$ over Bob views. □

6.4.3 General Definition

We introduce a more general definition of atomic sub-modular oracle

Definition 3 (Atomic Sub-modular Oracles)**.** *An oracle $\mathcal{O}$ is said to be a* atomic sub-modular oracle *if there exists a set of views* Unlikely *such that for any system interacting with the oracle with polynomial number of queries, the view that the system gets of the oracle is in* Unlikely *only with negligible probability, and the following holds. Consider any two query-answer sets $X, Y \subseteq Q$ for any $Q \notin$* Unlikely*, such that $|X^*|, |Y^*|$ are at most poly(n). Let q be a query not answered in $X^* \cup Y^*$ and a be any answer in the range of the oracle $\mathcal{O}$. Then it must hold that* $\Pr[\mathcal{O}(q) = a \mid X, Y] \approx_{(1+\nu)} \Pr[\mathcal{O}(q) = a \mid X]$.

This general definition is helpful in certain cases. Consider the case of a length tripling random oracles $\mathcal{R}$ and a test oracle Test which tells whether a point in the range has a pre-image or not. Suppose we restrict our attention to only injective random oracles $\mathcal{R}$. Then this pair of oracles satisfies this general definition. Similarly, we can consider the quartet of oracles (Gen, Enc, Test $_1$, Test $_2$) where both Gen and Enc are restricted to injective functions.

It is easy to see that the product characterization of Lemma 32 extends to this definition of oracles. This product characterization is crucially used to argue that the failure probability of Bounded-Eve$({n^2}/{\varepsilon^2}, \pi(x,y))$ is low. If the product characterization is satisfied only approximately, instead of exactly, then we can still show that the probability of failure of Bounded-Eve($^{n^2}/_{\varepsilon^2}, \pi(x,y)$) is low, though this bound is weaker than the case when the product characterization holds exactly. Thus, we can generality Theorem 6 to the following:

Theorem 8. *Suppose there exists a* $(1 - \mathit{negl})$ *semi-honest secure protocol for an undecomposable* $\mathcal{F}$ *when parties have access to a atomic sub-modular oracle and* PSPACE *oracle; then there exists a* $(1 - {}^{1}/_{\mathit{poly}})$ *semi-honest secure for* $\mathcal{F}$ *where parties have access to a* PSPACE *oracle and arbitrarily chosen polynomial poly.*

This result will be used by [MMP11] to show that the generic PKE oracle proposed by [GKM$^+$00] is useless for semi-honest deterministic SSFE.

Chapter 7

Conclusion

As exhibited by our results, the traditional study of triviality and completeness is not sufficient to fully understand the complexity of secure multi-party computation. Although these two aspects are good points to start our study of cryptographic complexity theory, there are several structures which are left unexplored. A sophisticated treatment of the topic based on a reduction based framework can exhibit significantly more structure and intricacies. As immediate extensions of the work presented in this dissertation, we mention some open problems.

7.1 Randomized Function Evaluation

In two-party symmetric randomized function evaluation, Alice and Bob obtain an output which is distribution which is a function of their local inputs. Our understanding of which two-party symmetric randomized function evaluations can be securely performed when parties are semi-honest corrupt is extremely limited. We are only aware of completeness result by [Kil00] and the problem of triviality is still open. Surprisingly, this problem is non-trivial even when Alice and Bob have binary inputs and there are at least three output alphabets. One of the main motives of studying the $\mathcal{F}_{\text{COIN}}$-hybrid in [MOPR11, MP11] is to gain intuition for this problem.

Open Problem 1. *Characterize semi-honest trivial two-party symmetric randomized function evaluations.*

7.2 Implication and Equivalences

We have seen that when we consider UC secure reductions, most reductions involving two party functionalities are either unconditionally true, intermediate to OWF assumption and $\mathsf{sh\text{-}OT}$ assumption or false. When we consider UC-security against parties with bounded computational power, we showed that several reductions $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ are equivalent to OWF assumption and $\mathsf{sh\text{-}OT}$ assumption. In fact it, could be possible that these reductions have intermediate complexity, but we conjecture that it is not the case.

Open Problem 2. *When we consider UC-secure realization of two-party functionalities against adversaries with bounded computational power, are all reductions of the form $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ equivalent to* OWF *assumption and* $\mathsf{sh\text{-}OT}$ *assumption.*

But, as shown in Chapter 6, if we change the notion of security, the number of intermediate assumptions could possibly be infinite.

7.3 Separations

We considered the separation of the assumption: $\mathcal{F} \sqsubseteq^{\text{PPT}} \emptyset$ for semi-honest security and $\mathcal{F}$ being a non-trivial incomplete two-party deterministic function evaluation. We showed that this assumption is separated from the assumptions that one-way functions exist and public-key encryption is possible. But it is unknown whether existence of semi-honest secure protocol for oblivious transfer is separated from such assumptions. The main technical hurdle here is to provide an oracle with limited power which helps compute a non-trvial incomplete two-party deterministic function but does not help perform oblivious transfer. This construction is extremely important to understand the "irreducibility vs. separation" conjecture:

Open Problem 3. *Construct an oracle which helps perform a nontrivial incomplete function evaluation but is insufficient to compute oblivious transfer.*

Consider secure function evaluation against semi-honest adversaries. It has been shown that there are $\mathcal{F}$ such that $\mathcal{F} \sqsubseteq^{\text{PPT}} \emptyset$ is separated from OWF assumption and PKE assumption [MMP11].

Open Problem 4. *Are there infinitely many distinct assumptions of the form $\mathcal{F} \sqsubseteq^{\text{PPT}} \emptyset$, where we consider semi-honest secure computation?*

Consider any two functionalities $\mathcal{F}$ and $\mathcal{G}$ such that $\mathcal{F}$ does not statistically reduce to $\mathcal{G}$, i.e. $\mathcal{F} \not\sqsubseteq^{\text{STAT}} \mathcal{G}$. We can consider different notions of security like semi-honest security, standalone security, UC-security etc. Can such an irreducibility result be transformed into a separation result? Observe that if $\mathcal{G}$ itself can be securely realized then the problem is trivial. The non-trviality of the problem arises for the case when $\mathcal{G}$ itself cannot be securely realized. The random oracle to show such a separation cannot be too powerful, other it would also help securely realize $\mathcal{F}$ and not just $\mathcal{G}$. On the other hand, an extremely weak random oracle will be useless to realize $\mathcal{G}$. So, proving such a separation necessitates appropriately providing tight characterization of the complexity of securely realizing $\mathcal{G}$ and enabling the oracle with power sufficient to securely compute $\mathcal{G}$.

Open Problem 5. *Consider $\mathcal{F}$ and $\mathcal{G}$ such that $\mathcal{F}$ does not statistically reduce to $\mathcal{G}$ for some notion of security and $\mathcal{G}$ itself is non-trivial function. Does there exist an oracle with respect to which $\mathcal{G}$ has a secure protocol but $\mathcal{F}$ does not?*

7.4 Coin Tossing

Recently, Haitner and Omri [HO11] have shown that secure strong coin-tossing protocols entail existence of one-way functions. But the corresponding problem for weak coin-tossing is still open. Moreover, exploring whether the attack presented in [HO11] is optimal or not is extremely important due to the reduction between strong and wek coin tossing by [CK09].

Open Problem 6. *Is existence of one-way function essential for secure weak coin-tossing protocols? Can we achieve beyond $1/\sqrt{2}$ bias in strong coin-tossing protocols if one-way functions do not exist?*

For general protocols, it was shown in [MPS10] that if $\mathsf{NP} \subseteq \mathsf{BPP}$, then one of the parties can force its preferred outcome with probability $3/4$. And this bound is optimal for any "local algorithm". But can a non-local algorithm overcome this bound?

Open Problem 7. *If $\mathsf{NP} \subseteq \mathsf{BPP}$, can one of the parties force it preferred outcomes with probability $> 3/4$. In fact is it possible that one party can force its preferred outcome with near certainty.*

Appendix A

Implications and Equivalences

A.1 Details for the Proof of Theorem 5

In this section we complete the proof of Theorem 5 outlined in the main body. What remains to be shown are the details of Lemma 15 — namely, that if the dimensions of one exchange function $\mathcal{F}$ are not "smaller" than the dimensions of another exchange function $\mathcal{G}$, then $\mathcal{F} \sqsubseteq^{\text{PPT}} \mathcal{G}$ implies the $\mathsf{sh\text{-}OT}$ assumption.

We develop the proof in several parts. First, to introduce our approach to proving separations involving exchange functions, we show that $\mathcal{F}^{2,2}_{\text{EXCH}} \sqsubseteq^{\text{PPT}} \mathcal{F}_{\text{COIN}}$ implies $\mathsf{sh\text{-}OT}$ assumption. Then we show that $\mathcal{F}^{i,j}_{\text{EXCH}} \sqsubseteq^{\text{PPT}} \mathcal{F}^{(i-1),(j-1)}_{\text{EXCH}}$ implies the $\mathsf{sh\text{-}OT}$ assumption, and finally that $\mathcal{F}^{i,j}_{\text{EXCH}} \sqsubseteq^{\text{PPT}} \mathcal{F}^{i',j'}_{\text{EXCH}}$ implies the $\mathsf{sh\text{-}OT}$ assumption, where $\min\{i', j'\} < i, j \leq \max\{i', j'\}$. These two cases suffice to prove the desired characterization.

A.1.1 Reduction to $\mathcal{F}_{\text{COIN}}$.

Lemma 33. $\mathcal{F}^{2,2}_{\text{EXCH}} \sqsubseteq^{\text{PPT}} \mathcal{F}_{\text{COIN}}$ *implies the* $\mathsf{sh\text{-}OT}$ *assumption.*

Proof. Let π be a secure protocol for $\mathcal{F}^{2,2}_{\text{EXCH}}$ in the $\mathcal{F}_{\text{COIN}}$-hybrid world. We will transform π to obtain a secure protocol for $\mathcal{F}_{\text{OT}}$ against semi-honest adversaries.

Let s_B be the random variable denoting the round in which the simulator extracts from a passively corrupt Alice and sends her input to $\mathcal{F}^{2,2}_{\text{EXCH}}$ and T_A be the round where any adversary with unbounded computational power can guess Bob's input with probability at least $\zeta = 3/4$ given Alice's view. Fix any passive adversarial strategy for Alice which outputs a guess of Bob's input at each step of the protocol, and define t_A as the random variable denoting the round when this guess is correct with probability at least $\zeta = 3/4$ (where the probability is over the randomness independent of Alice's view), *when interacting with the simulator.* It is easy to see that $t_A \geq T_A$. First, we observe that we can assume, without loss of generality, that the simulator always extracts in a round where Alice sends a message to the simulator. Otherwise, if the simulator extracts in a round where she sends a message to Alice or after receiving a request to run $\mathcal{F}_{\text{COIN}}$, we can construct another simulator which could have extracted one round earlier. By the definition of the simulation, Alice's view is completely independent of Bob's input through the first s_B rounds (even

Protocol for a weak variant of $\mathcal{F}_{\text{OT}}$. Alice has inputs $x_0, x_1 \in \{0, 1\}$, and Bob has input $b \in \{0, 1\}$.

1. Alice runs two instances of the protocol π with Bob, using inputs x_0 and x_1, respectively.
2. Bob picks a random $r \in [r(k)]$, where $r(k)$ is a polynomial bound on the number of rounds in π.
3. In the bth instance of π, Bob runs the simulator for π against Alice (including simulating her interface with instances of $\mathcal{F}_{\text{COIN}}$), and halts the interaction after the rth round of π.
4. In the $(1 - b)$ instance of π, Bob runs the π protocol honestly with Alice on a fixed input (say, 0), and also honestly simulates all instances of $\mathcal{F}_{\text{COIN}}$ for Alice. Bob halts the interaction after the rth round of π.
5. If the simulator has extracted x_b, then Bob outputs it. Otherwise, he asks Alice for (x_0, x_1), and she sends it to him.

Figure A.1: Weak oblivious transfer protocol, using any secure protocol π for $\mathcal{F}^{2,2}_{\text{EXCH}}$ in the $\mathcal{F}_{\text{COIN}}$-hybrid world.

in the presence of an ideal $\mathcal{F}_{\text{COIN}}$). Thus $t_A \geq T_A \geq s_B + 1$, and in particular, $E[t_A] \geq E[s_B] + 1$.

Now consider running this passive adversarial strategy for Alice against an honest Bob in the actual protocol execution, instead of against the simulator. We define u_A to be the random variable denoting the first round in which Alice's guess is correct with probability at least ζ. By the security of π, these two interactions must be indistinguishable to this Alice strategy, thus $|E[u_A] - E[t_A]| < \varepsilon/\zeta = \varepsilon'$, where ε is the negligible simulation error of the protocol. Thus $E[u_A] \geq E[s_B] + 1 - \varepsilon'$.

Similarly we can define u_B and s_A and conclude that $E[u_B] \geq E[s_A] + 1 - \varepsilon'$. Then, either $E[u_A] \geq E[s_A] + 1 - \varepsilon'$, or $E[u_B] \geq E[s_B] + 1 - \varepsilon'$; otherwise we would get that $E[u_A] < E[s_A] + 1 - \varepsilon' \leq E[u_B] < E[s_B] + 1 - \varepsilon'$.

By symmetry, we assume that $E[u_B] \geq E[s_B] + 1 - \varepsilon'$. In other words, in an interaction with an honest Alice, the simulator will, on average, extract Alice's input earlier than any passive Bob could guess Alice's input with probability at least ζ.

Now consider the protocol given in Figure A.1. First, since Alice cannot distinguish a simulated instance of π from an honest execution of π, Alice has no advanatage in predicting Bob's bit b. Thus the protocol gives complete privacy for Bob.

Then a passively corrupt Bob in this protocol can guess Alice's input x_{1-b} correctly with probability at most

$$\begin{aligned}
&\Pr[s_B \leq r < u_B]\zeta + (1 - \Pr[s_B \leq r < u_B]) \\
&= 1 - (1 - \zeta)\Pr[s_B \leq r < u_B] \\
&\leq 1 - (1 - \zeta)E[u_B - s_B]/r(k) \\
&\leq 1 - (1 - \zeta)(1 - \varepsilon')/r(k)
\end{aligned}$$

by the definition of u_B. Or, in other words, Bob's guess is incorrect with probability at least $(1-\zeta)(1-\varepsilon')/r(k)$, which is an inverse polynomial in the security parameter. In Appendix A.2, we show how a weak $\mathcal{F}_{\text{OT}}$ protocol with this security property can be amplified to give a full-fledged (semi-honest) $\mathcal{F}_{\text{OT}}$ protocol. □

A.1.2 Reductions Between Exchange Functions

We first establish a convenient technical lemma:

Lemma 34. *For each $j \in [i]$, let $\mathcal{D}_j$ be a probability distribution over the elements $\{m_1, \ldots, m_{i-1}\}$. Now consider the following experiment: Choose $j \in [i]$ in random, and then output a sample according to $\mathcal{D}_j$.*

The probability of correctly predicting j given only the output of this procedure is at most $(i-1)/i$.

Proof. Let $p_{u,v}$ be the probability of sampling message m_v when using $\mathcal{D}_u$. So, we have:

$$\sum_{v=1}^{i-1} p_{u,v} = 1 \text{ for all } u \in [i]$$

Let $q_{v,u}$ be the probability of outputting u after seeing message v. So, we have:

$$\sum_{u=1}^{i} q_{v,u} = 1 \text{ for all } v \in [i-1]$$

The probability of being correct is:

$$\begin{aligned}
\zeta &= \frac{\sum_{u=1}^{i}\sum_{v=1}^{i-1} p_{u,v}q_{v,u}}{i} \\
&= \frac{\sum_{v=1}^{i-1}\sum_{u=1}^{i} p_{u,v}q_{v,u}}{i} \\
&\leq \frac{\sum_{v=1}^{i-1} \max_{u=1}^{i} p_{u,v}}{i} \\
&\leq \frac{i-1}{i}
\end{aligned}$$

□

Before we prove the general result, let us prove an intermediate result

Lemma 35. *Let $i \geq 3$. Then $\mathcal{F}_{\text{EXCH}}^{i,i} \sqsubseteq^{\text{PPT}} \mathcal{F}_{\text{EXCH}}^{(i-1),(i-1)}$ implies the* $\mathsf{sh\text{-}OT}$ *assumption.*

Proof. The proof is very similar to that of the previous lemma. However, now that the purported protocol π can use $\mathcal{F}_{\text{EXCH}}^{(i-1),(i-1)}$, we must consider information about the parties' inputs that is exchanged via the ideal functionality.

Note that in the proof of the previous lemma, we would obtain a suitable weak OT protocol (i.e., amenable to amplification) even if ζ is at most $1 - \frac{1}{\text{poly}}$ in the security parameter

Consider $\zeta = c + \frac{i-1}{i}$, where $c > 0$ is any constant. As before, we let s_B be the round during which the simulator extracts from a passively corrupt Alice. Thus, Alice may send an input to her interface of $\mathcal{F}_{\text{EXCH}}^{(i-1),(i-1)}$, then the simulator will send the extracted input to $\mathcal{F}_{\text{EXCH}}^{i,i}$, receive the output, and then complete the round by simulate the response of the simulated $\mathcal{F}_{\text{EXCH}}^{(i-1),(i-1)}$ functionality to Alice.

The simulator will complete the round by simulating a response from $\mathcal{F}_{\text{EXCH}}^{(i-1),(i-1)}$, which will be an element of $[i-1]$. At the start of round s_B, Alice's view is independent of the honest Bob's input $y \in [i]$ to $\mathcal{F}_{\text{EXCH}}^{i,i}$. There are only $i-1$ possible responses the simulator can provide after receiving y from the ideal $\mathcal{F}_{\text{EXCH}}^{i,i}$ functionality. So after round s_B is complete, Alice cannot guess y with probability greater than $(i-1)/i < \zeta$, where $\nu(\cdot)$ is negligible. Since t_B is defined as the first point at which Alice can guess Bob's input with probability at least ζ, we have $t_B \geq s_A + 1$.

Similarly we can conclude that $t_A \geq s_B + 1$. Rest of the proof is identical to the proof mentioned above. □

Finally we move to our general result.

Lemma 36. *Let i, j, i', j' be such that ($i > i'$ or $j > j'$) and ($i > j'$ or $j > i'$). Then $\mathcal{F}_{\text{EXCH}}^{i,j} \sqsubseteq^{\text{PPT}} \mathcal{F}_{\text{EXCH}}^{i',j'}$ implies the* sh-OT *assumption.*

Proof. The proof is very similar to that of the previous lemma. However, now that the purported protocol π can use $\mathcal{F}_{\text{EXCH}}^{i',j'}$, we must consider information about the parties' inputs that is exchanged via the ideal functionality.

Note that in the proof of the previous lemma, we would obtain a suitable weak OT protocol (i.e., amenable to amplification) even if ζ is $1 - \frac{1}{\text{poly}}$ in the security parameter, and one of $\{E[t_B - s_B], E[t_A - s_A]\}$ is at least $\frac{1}{\text{poly}}$ in the security parameter.

Case 1: ($\max\{i,j\} > \max\{i',j'\}$): Suppose $i \geq j$ and $i > i' \geq j'$ and Bob feeds input from $[i]$ into the ideal functionality. We define $\zeta = c + \frac{i-1}{i}$. Now we define s_B and t_A as we had done earlier. Similar to the argument in the previous lemma we get that $t_A \geq s_B + 1$ (because $i - 1 \geq i' \geq j'$). It is always the case that $t_B \geq s_A$. So, we get the condition that $t_A \geq s_B + 1$ and $t_B \geq s_A$.

In general we can say that:

$$(t_A \geq s_B \text{ and } t_B \geq s_A + 1), \text{ or}$$

$$(t_B \geq s_A \text{ and } t_A \geq s_B + 1)$$

These conditions imply that:

$$E[u_A] \geq E[s_A] + ({}^1\!/\!{}_2 - \varepsilon'), \text{ or}$$
$$E[u_B] \geq E[s_B] + ({}^1\!/\!{}_2 - \varepsilon')$$

Observe that in our weak OT construction, all we needed was that $E[u_A - s_A]$ or $E[u_B - s_B]$ is $\frac{1}{\text{poly}}$ in the security parameter. So, we can continue with our weak OT construction as we had mentioned earlier.

Case 2: $(\min\{i', j'\} < i, j \leq \max i', j')$: Observe that even if for some polynomial $\lambda(\cdot)$ we have:

$$\left(E[t_A] \geq E[s_B] \text{ and } E[t_B] \geq E[s_A] + \frac{1}{\lambda(k)}\right), \text{ or}$$
$$\left(E[t_B] \geq E[s_A] \text{ and } E[t_A] \geq E[s_B] + \frac{1}{\lambda(k)}\right)$$

we can use the approach mentioned above to get the weak OT protocol. So, we just need to consider the case when $E[t_B] \in \left[E[s_A], E[s_A] + \frac{1}{\lambda(k)}\right)$ and $E[t_A] \in \left[E[s_B], E[s_B] + \frac{1}{\lambda(k)}\right)$, where $\lambda(\cdot)$ is a suitably chosen large polynomial.

In this case, we will prove that:

1. $\Pr(t_B \geq s_B + 1)$ or $\Pr(t_A \geq s_A + 1)$ is $\geq \frac{1}{5}$

2. $|\Pr(u_A = i) - \Pr(t_A = i)|$, $|\Pr(u_B = i) - \Pr(t_B = i)|$ are both $\leq \frac{1}{\rho(k)}$ for any polynomial ρ

These will imply that our weak OT construction will work in this case as well.

Now, we show that the above mentioned properties hold. If $E[t_B] \in \left[E[s_A], E[s_A] + \frac{1}{\lambda(k)}\right)$ and $E[t_A] \in \left[E[s_B], E[s_B] + \frac{1}{\lambda(k)}\right)$, then with probability $\geq 1 - \frac{2}{\lambda(k)n}$ we will have the event that $t_B = s_A$ and $t_A = s_B$. Consider the set of rounds S where $t_A = s_B$. Similarly define T to be the set of rounds where $t_B = s_A$. WLOG, we can assume that Alice uses i' side of $\mathcal{F}_{\text{EXCH}}^{i',j'}$ only in even rounds and the j' side of the $\mathcal{F}_{\text{EXCH}}^{i',j'}$ only in odd rounds. So, we conclude that the sets S and T are mutually disjoint.

Let $x_S(i)$ be the probability of the event $t_A = s_B \leq i$ happens. Similarly define $x_T(i)$ as the probability of the event $t_B = s_A \leq i$ happens. Initially $x_S(0) = x_T(0) = 0$ and $x_S(n) = x_T(n) = 1 - \frac{2}{\lambda(k)n}$. So look at the first i such that $x_S(i)$ or $x_T(i)$ becomes $\geq \frac{1}{2}\left(1 - \frac{2}{\lambda(k)n}\right)$. Observe that at any given round only $x_S(i)$ or $x_T(i)$ changes. WLOG assume that $x_S(i)$ reaches the threshold first. Then since $y_S(i)$ could not have changed at this round, we get that $y_S(i) \leq \frac{1}{2}\left(1 - \frac{2}{\lambda(k)n}\right)$. Then we see that with probability $\geq \left({}^1\!/\!{}_2 - \frac{1}{\lambda(k)n}\right)^2 \geq \frac{1}{4} - \frac{2}{\lambda(k)n} \geq \frac{1}{5}$, we have the event that $s_B \leq t_B - 1$.

Now, all we need to show is that $\Pr(u_B = i)$ and $\Pr(t_B = i)$ are 1/poly-close. We pick a suitable polynomial ρ. We run an honest execution of the protocol against a simulator for Alice. We can estimate

$\Pr(t_B = i)$ within $\frac{1}{\rho}$ additive error in polynomial time. Similarly, we run an honest execution of the protocol against honest Alice. We can estimate $\Pr(u_B = i)$ within $\frac{1}{\rho}$ additive error in polynomial time.

If $|\Pr(t_B = i) - \Pr(u_B = i)| > \frac{3}{\rho}$, then we can create a polynomial time distinguisher which distinguishes between the real and ideal world. So, for every round $i \in [r(k)]$, $|\Pr(t_B = i) - \Pr(u_B = i)| \leq \frac{3}{\rho}$.

Given the guarantee that, for all $i \in [r(k)]$, $|\Pr(t_B = i) - \Pr(u_B = i)| \leq \frac{3}{\rho}$ and $\Pr(s_B \leq t_B - 1) \geq \frac{1}{5}$, the construction given earlier gives us a weak OT. □

A.2 Oblivious Transfer Amplification

We first establish the following convenient technical lemma:

Lemma 37 (Noisy Channel Bounds). *Consider a noisy channel $\mathcal{C}$, which either forwards an input element $x \in \mathbb{Z}_N$ unchanged with probability q, and otherwise replaces it uniformly chosen element from $\mathbb{Z}_N \setminus \{x\}$.*

Suppose a string $s = s_1 \dots s_k \in \mathbb{Z}_N^k$ is passed through $\mathcal{C}$, and $t = t_1 \dots t_k$ is the result. Then the probability that $\sum_{i=1}^k t_i = \sum_{i=1}^k s_i$ is at most

$$\frac{1}{N} + \exp\left(-\frac{1}{N} - \frac{(1-q)k}{(N-1)}\right)$$

.

Proof. Without loss of generality, suppose that $\sum_{i=1}^k s_i = 0$. Consider the following polynomial:

$$f(x) = \left(q + \frac{1-q}{N-1}x + \dots \frac{1-q}{N-1}x^{N-1}\right)^k$$

Observe that the probability that $\sum_{i=1}^k t_i = 0$ is given by the following expression:

$$\sum_{\lambda \in \mathbb{Z}} [x^{\lambda N}] f(x) = \frac{\sum_{i=0}^{n-1} f(\omega^i)}{N},$$

where $1, \omega, \ldots, \omega^{N-1}$ are distinct roots of $z^N = 1$. We can evaluate the expression in the following manner:

$$\begin{aligned}\frac{1}{N}\sum_{i=0}^{N-1} f(\omega^i) &= \frac{1}{N}\sum_{i=0}^{N-1}\left(\frac{Nq-1}{N-1} + \frac{1-q}{N-1}\sum_{j=0}^{N-1}\omega^{ij}\right)^k \\ &= \frac{1}{N} + \frac{(N-1)\left(\frac{Nq-1}{N-1}\right)^k}{N} \\ &= \frac{1}{N} + \left(1-\frac{1}{N}\right)\left(1-\frac{1-q}{N-1}\right)^k \\ &\le \frac{1}{N} + \exp\left(-\frac{1}{N} - \frac{(1-q)k}{(N-1)}\right) \qquad \square\end{aligned}$$

We now define our variant of a weak Oblivious Transfer (OT) and how it can be amplified to obtain the conventional one-out-of-two OT.

Similar to the definition of (p, q)-OT used in [DKS99], we introduce a notion of a weak OT.

Definition 4 (q-weak-OT)**.** *A q-weak-OT is a protocol that satisfies the following conditions:*

- *The sender has inputs $(x_0, x_1) \in \mathbb{Z}_N^2$. The receiver has input $b \in \{0, 1\}$ to the functionality and receives x_b as output.*

- *A passively corrupt sender has no advantage in guessing the bit b.*

- *No passively corrupt receiver can guess x_{1-b} with probability greater than q, when the sender's inputs are random.*

Thus, $\frac{1}{N}$-weak-OT is a standard OT with sender input set $\mathbb{Z}_N$.

We can amplify a q-weak-OT using an algorithm taken from [DKS99].

Definition 5 (R-Reduce)**.** *R-Reduce$(k, \mathcal{W})$ is defined as the following protocol, where $\mathcal{W}$ is a weak-OT.*

1. *Let $(x_0, x_1) \in \mathbb{Z}_N^2$ be the input of the sender; and $b \in \{0, 1\}$ be the input of the receiver.*

2. *The sender generates random $(x_{0i}, x_{1i}) \in \mathbb{Z}_N^2$, for $i \in [k]$. Let $r_0 = \sum_{i=1}^k x_{0i}$ and $r_1 = \sum_{i=1}^k x_{1i}$. The sender sends $z_0 = x_0 + r_0$ and $z_1 = x_1 + r_1$ to the receiver*

3. *Both parties execute $\mathcal{W}$, k times with input $(x_{0i}, x_{1i}) \in \mathbb{Z}_N^2$ for the sender and input b for the receiver.*

4. *The receiver outputs $x_b = z_b - (\sum_{i=1}^k x_{b,i})$.*

Lemma 38. *If $\mathcal{W}$ is a q-weak-OT, then R-Reduce$(k, \mathcal{W})$ is a $(\frac{1}{N} + \nu(q, k))$-weak-OT, where:*

$$\nu(q, k) \le \exp\left(-\frac{1}{N} - \frac{(1-q)k}{(N-1)}\right)$$

Proof. We consider the probability that the receiver can successfully guess x_{1-b}. Let $s = s_1 \ldots s_k \in \mathbb{Z}_N^k$ be chosen uniformly at random.Suppose we are given a string $t_1 \ldots t_k \in \mathbb{Z}_N^k$ which has the property that $t_i = s_i$ with probability q. Observe that if t_i is wrong, it adds an error $s_i - t_i$ which is uniformly random over $\mathbb{Z}_N$. So, in general with probability q it either adds 0 error; or adds a random error from the set $\mathbb{Z}_N \setminus \{0\}$ with probability $(1-q)/(N-1)$. Then, using Lemma 37, the probability that $\sum_{i=1}^k s_i = \sum_{i=1}^k t_i$ is at most:

$$\frac{1}{N} + \exp\left(-\frac{1}{N} - \frac{(1-q)k}{(N-1)}\right) \qquad \square$$

Thus, if $q \leq 1 - \frac{1}{\mathrm{poly}(k)}$, then R-Reduce$(\kappa/(1-q), \mathcal{W})$ is a full-fledged 1-out-of-2 OT protocol.

Appendix B

Weak Coin

B.1 Examples

Greedy does not work well. Greedy strategy is one of the basic strategies to bias the outcome towards b. At a partial transcript v, output a bit d such that the color χ_{vd} is closer to b than $\chi_{v(1-d)}$. But this strategy is not good and we explicitly construct a protocol tree where we can make the bias obtained by the greedy algorithm arbitrarily small. Recall that if A/χ_v is written at a node, it means that Alice is supposed to send the next message after the partial transcript v is generated and the subtree S_v computes a χ_v-coin when both parties are honest. For simplicity, when we explain the transcript tree construction, we do not insist that Alice and Bob nodes alternate. If an Alice node v follows an Alice node v', then we can assume that there is a dummy Bob node v'' such that whatever bit is sent at v'' it does not not effect the outcome. We can assume that both children of v'' are identical to v'.

Consider the following recursive graph construction (Figure B.1):

1. For the base case of $k = 0$, define G_0 as the tree where Alice announces the outcome of the $1/2$-coin.

2. For any other $k > 0$, we recursively define G_k using G_{k-1}. The root node v is an Alice node that implements a $1/2$-coin. Its two children $v0$ and $v1$ are Bob nodes which implement $1/2 - \varepsilon$ and $1/2 + \varepsilon$-coins respectively. Nodes $v00$ is an Alice node implementing $1/2 - 2\varepsilon$-coin and $v11$ is a Bob node implementing $1/2 + 2\varepsilon$ coin. The S_{v01} and S_{v10} are the tree G_{k-1}.

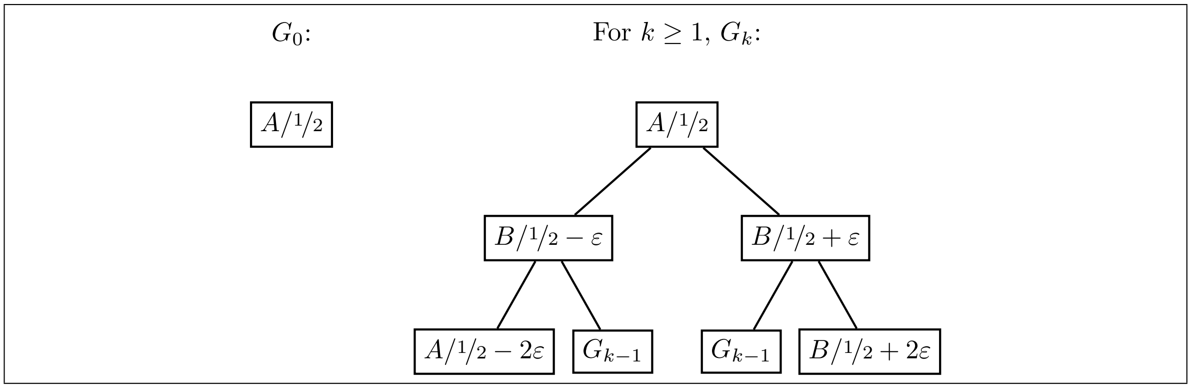

Figure B.1: Greedy strategy is not good.

Let $g_b^A(k)$ (resp. $g_b^B(k)$) be the expectation of the outcome when Alice (resp. Bob) is trying to bias the outcome to b in G_k.

$$\begin{aligned} g_1^A(k) &= 1/2 g_1^A(k-1) + \left(\frac{1}{4} + \varepsilon\right) = 1/2 + 2\varepsilon + \frac{\Theta(1)}{2^k} \\ g_0^B(k) &= 1/2 g_0^B(k-1) + \left(\frac{1}{4} - \varepsilon\right) = 1/2 - 2\varepsilon + \frac{\Theta(1)}{2^k} \end{aligned}$$

By symmetry, $1 - g_0^A(k) = g_1^A(k)$ and $1 - g_1^B(k) = g_0^B(k)$. We see that we can drive the bias obtained by the greedy algorithm to negligibly close to $1/2$.

Let $h_b^A(k)$ (resp. $h_b^B(k)$) be the expectation of the outcome when Alice (resp. Bob) is trying to bias the outcome to b in G_k. We can write the following recurrence:

$$\begin{aligned} h_1^A(k) &= 1/2 h_1^A(k-1) + \left(\frac{3}{8} + \frac{2\varepsilon^2(1-\varepsilon)}{(1+4\varepsilon^2)}\right) \left(\mathsf{Adv}_A^{(1)} \text{ on } G_k\right) \\ &= \frac{3}{4} + \frac{4\varepsilon^2(1-\varepsilon)}{(1+4\varepsilon^2)} + \frac{\Theta(1)}{2^k} \\ h_0^B(k) &= 1/2 h_0^B(k-1) + \left(\frac{1}{8} - 2\varepsilon^2\right) \left(\mathsf{Adv}_B^{(0)} \text{ on } G_k\right) \\ &= \frac{1}{4} - 4\varepsilon^2 + \frac{\Theta(1)}{2^k} \end{aligned}$$

This shows that our attack achieves nearly close to 3/4 and 1/4 bias.

Need to attack at more than constant rounds. Consider the recursive graph construction as shown in Figure B.2. Fix a particular k and consider the tree G_k such that the probability that the honest protocol reaches any $A/1/2$ leaf is ε and the probability of reaching any $B/1/2$ leaf is $\frac{1}{k+1} - \frac{k\varepsilon}{k+1}$. If Bob is honest then Alice can not bias the outcome by more than $k\varepsilon$, which can be made arbitrarily small. If Bob is not honest and he attacks at only c rounds, then the maximum bias he could generate is $\frac{c}{k+1} - \frac{kc\varepsilon}{k+1}$. So, to go beyond

1/poly(k) bias, Bob needs to attack at more than constant number of rounds.

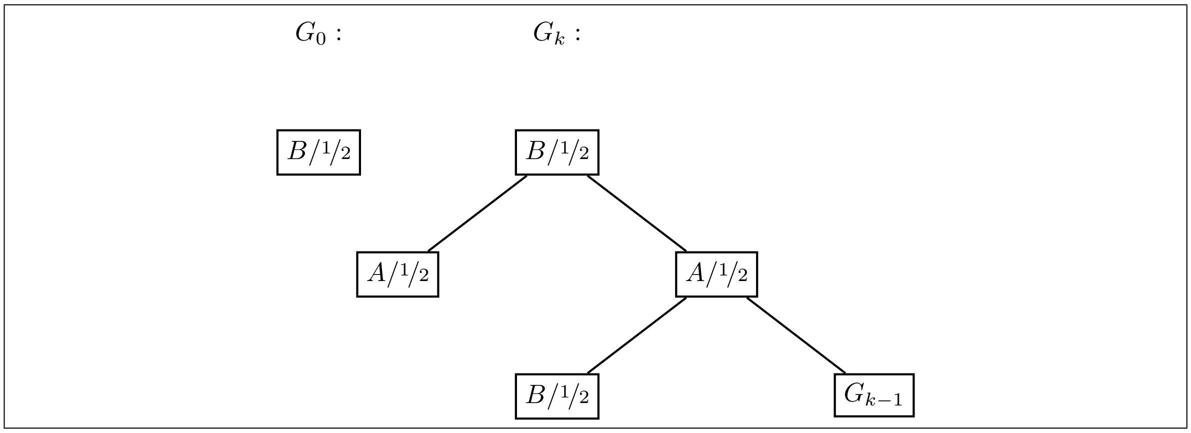

Figure B.2: Attacking constant number of grounds does not help.

Beyond 1/4 **bias.** Consider the class of adversarial strategies which uses only χ_v, χ_{v0}, χ_{v1} and $|v|$ to determine the distribution according to which the next bit is sampled. We will show that any adversary in this class can not bias by more than 1/4. Consider the performance of such adversaries on the graph in Figure B.3. Suppose the probability of reaching any child vb from a node v in the honest protocol is ${}^1\!/\!_2$. If Bob is not honest, then with probability ${}^1\!/\!_2$it can decide the outcome of a ${}^1\!/\!_2$-coin. So, the maximum bias it can obtain is 1/4. Now, if Alice is not honest then she can reach a leaf $A/{}^1\!/\!_2$ coin with probability ${}^1\!/\!_2$independent of her strategy. So, she can obtain a maximum bias of 1/4.

If we expand the class of adversaries to include any adversary with constant look ahead in the protocol tree, then we can generalize the graph in Figure B.3 so that they can obtain at most 1/4 bias. The class of adversaries we have considered currently can be interpreted as 1-look ahead adversaries. Suppose, we introduce c redundant levels between any two levels of the graph in Figure B.3. Then any adversary with $(c+1)$ look ahead in the protocol tree can not obtain more than 1/4 bias. Note that by adding dummy nodes, even if the adversary's strategy looks ahead a bounded depth or tries to take into account whose turn is next, this cannot help it achieve a better bias.

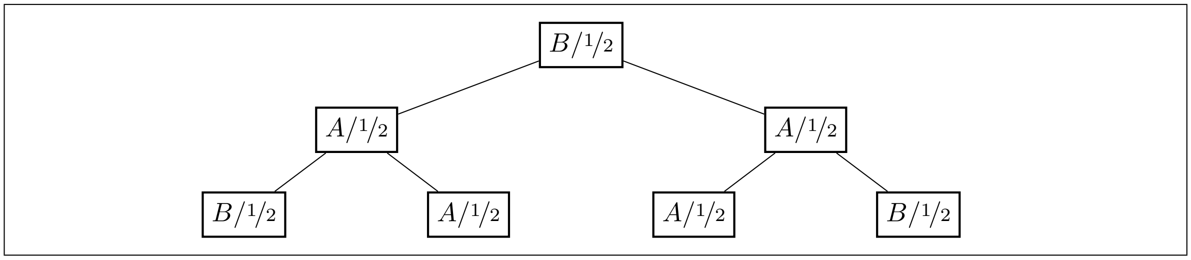

Figure B.3: Certain class of local algorithms can not go beyond 1/4 bias.

B.2 Deferred Calculations

Lemma 39. *If $\chi_0, \chi_1, p_0 \in (0,1)$ then $1 - r^*_{(1-c)} < r^*_c < p_c$.*

Proof. Recall that $r^*_{(1-c)} = p_{(1-c)} \left[\frac{\chi_{high} - \chi\chi_{high}}{\chi - \chi\chi_{high}}\right]$ and $r^*_c = p_c \left[\frac{\chi_{low} - \chi\chi_{low}}{\chi - \chi\chi_{low}}\right] = p_c \frac{\left(\frac{1}{\chi} - 1\right)}{\left(\frac{1}{\chi_{low}} - 1\right)} < p_c$, where $\chi = p_c\chi_{low} + p_{(1-c)}\chi_{high}$ and $p_c + p_{(1-c)} = 1$.

$$\begin{aligned}
1 - r^*_{(1-c)} &= 1 - \frac{p_{(1-c)}(1-\chi)\chi_{high}}{(1-\chi_{high})\chi} \\
&= \frac{(\chi - p_{(1-c)}\chi_{high}) - (\chi\chi_{high} - p_{(1-c)}\chi\chi_{high})}{(1-\chi_{high})\chi} \\
&= p_c \left[\frac{\chi_{low} - \chi\chi_{high}}{\chi - \chi\chi_{high}}\right]
\end{aligned}$$

Consider the function $f(x) = p_c \left[1 - \frac{\chi - \chi_{low}}{\chi - \chi x}\right]$. It is easy to see that it is a monotonically decreasing function. Observe that $f(\chi_{low}) = r^*_c$ and $f(\chi_{high}) = 1 - r^*_{(1-c)}$. So, $1 - r^*_{(1-c)} < r^*_c < p_c$. □

Lemma 40. *Let $\chi = p_0\chi_0 + p_1\chi_1 \leq 1 - \delta$ and $p_0, p_1 \in [0,1]$ such that $p_0 + p_1 = 1$. Suppose $\tilde{r}_0 = r_0 + e$ and $\tilde{r}_1 = 1 - \tilde{r}_0$, where $r_c \leq p_c$ and $e \in [-B, B]$, then:*

$$\left(\frac{\tilde{r}_0(1-\chi_0) + \tilde{r}_1(1-\chi_1)}{(1-\chi)}\right) \leq 1 + \frac{B}{\delta}$$

Proof. Consider the following manipulation:

$$\begin{aligned}
\left(\frac{\tilde{r}_0(1-\chi_0) + \tilde{r}_1(1-\chi_1)}{(1-\chi)}\right) &= \frac{1 - (r_c\chi_c + (1-r_c)\chi_{(1-c)}) + e(\chi_{(1-c)} - \chi_c)}{(1-\chi)} \\
&\leq \frac{1 - \chi + e}{(1-\chi)} \leq 1 + \frac{B}{\delta}
\end{aligned}$$ □

Lemma 41 (Case 0)**.** *If $\chi \geq 1 - (\delta + 2\lambda)$ or $\chi \leq (\delta + 2\lambda)$ and we honestly follow the protocol then $E \leq 2 \leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \nu_{h+1}$.*

Proof. The result is immediate from the observation that $E \leq 2$ and from the following inequality:

$$1 \leq \frac{\delta}{1 - (1 - (\delta + 2\lambda))} + \nu_1$$ □

Lemma 42 (Case 1)**.** *If $\chi - \chi_c \leq \lambda^{1/3} + 4\lambda$, $\chi \leq 1 - \delta$, and we substitute $\tilde{r}_0 = \frac{p_0\chi_0}{\chi}$ and $\tilde{r}_1 = \frac{p_1\chi_1}{\chi}$, then $E \leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \nu_{h+1}$.*

Proof. Since $\tilde{r}_c = \frac{p_c \chi_c}{\chi} > r_c^*$ and $\tilde{r}_{(1-c)} = \frac{p_{(1-c)}\chi_{(1-c)}}{\chi} < r_{(1-c)}^*$, we have $\tilde{T}_c < 0$ and $\tilde{T}_{(1-c)} > 0$. We know that $\frac{1}{\chi'} \leq \frac{1}{\chi}$ if and only if $\tilde{r}_c \leq h_c$. Because $e = 0$, we can use the bound in Lemma 40 in our lower bound to obtain:

$$\begin{aligned} E &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \frac{p_c \chi_c}{\chi}\left(\frac{\chi - \chi_c}{(1-\chi)}\right) + \nu_h' \\ &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \frac{\lambda^{1/3} + 4\lambda}{\delta} + \nu_h \\ &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \nu_{h+1} \end{aligned}$$

□

Lemma 43 (Case 2)**.** *If $\chi \leq 1 - \delta$, and we substitute $\tilde{r}_0 = t_0 + e$ and $\tilde{r}_1 = t_1 - e$, where $e \in [-9\lambda^{1/3}, 9\lambda^{1/3}]$ and $\tilde{r}_c \leq h_c$, then $E \leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \nu_{h+1}$.*

Proof. Recall that $\frac{1}{\chi'} \leq \frac{1}{\chi}$ if and only if $\tilde{r}_c \leq h_c$. Since $t_c < p_c$, we can use the bound in Lemma 40. When we substitute $\tilde{r}_0$ and $\tilde{r}_1$ and we get $\tilde{T}_{low} \geq 0$ and $\tilde{T}_{high} \geq 0$ then:

$$\begin{aligned} E^{(+,+)} &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \nu_h' \\ &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \nu_h\left(1 + \frac{9\lambda^{1/3}}{\delta}\right) \\ &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \nu_{h+1} \end{aligned}$$

If we substitute $\tilde{r}_0$ and $\tilde{r}_1$ and we get $\tilde{T}_0 \geq 0$ and $\tilde{T}_1 < 0$ then we know that $\tilde{r}_1 = r_1^* + e'$ such that $e' \in [0, 9\lambda^{1/3}]$ (Lemma 44). In this case:

$$\begin{aligned} E^{(+,-)} &\leq 1 + \frac{\delta}{(1-\delta)} + \frac{\delta}{\chi'} + \frac{e'(1-\chi_1)}{(1-\chi)} + \nu_h' \\ &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \frac{e'(1-\chi_1)}{(1-\chi)} + \nu_h\left(1 + \frac{9\lambda^{1/3}}{\delta}\right) \\ &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \frac{9\lambda^{1/3}}{\delta} + \nu_h\left(1 + \frac{9\lambda^{1/3}}{\delta}\right) \\ &= 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \nu_{h+1} \end{aligned}$$

Similarly, if $\tilde{T}_0 < 0$ and $\tilde{T}_1 \geq 0$ then:

$$\begin{aligned}
E^{(-,+)} &\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi'} + \frac{e'(1-\chi_0)}{(1-\chi)} + \nu'_h \\
&\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \frac{e'(1-\chi_0)}{(1-\chi)} + \nu_h \left(1 + \frac{9\lambda^{1/3}}{\delta}\right) \\
&\leq 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \frac{9\lambda^{1/3}}{\delta} + \nu_h \left(1 + \frac{9\lambda^{1/3}}{\delta}\right) \\
&= 1 + \frac{\delta}{(1-\chi)} + \frac{\delta}{\chi} + \nu_{h+1}
\end{aligned}$$

□

Lemma 44. *$t_c \leq r^*_c \leq h_c$ and $t_{(1-c)} \leq r^*_{(1-c)}$ and $t_c + t_{(1-c)} = 1$.*

Proof. It is trivial to see that $r^*_c = \frac{p_c\chi_c(1-\chi)}{\chi(1-\chi_c)} \leq h_c = \frac{p_c\chi_c}{\chi}$. The remainder of the proof is immediate from simple manipulation of terms:

$$\begin{aligned}
& t_c = \frac{p_c\chi_c(1-\chi_{(1-c)})}{(\chi - \chi_0\chi_1)} \leq \frac{p_c\chi_c(1-\chi)}{\chi(1-\chi_c)} = r^*_c \\
\iff \quad & \chi - (\chi_0 + \chi_1)\chi + \chi\chi_0\chi_1 \leq \chi - \chi_0\chi_1 - \chi^2 + \chi\chi_0\chi_1 \\
\iff \quad & 0 \leq (\chi_0 - \chi)(\chi - \chi_1) \\
\iff \quad & t_{(1-c)} = \frac{p_{(1-c)}\chi_{(1-c)}(1-\chi_c)}{(\chi - \chi_0\chi_1)} \leq \frac{p_{(1-c)}\chi_{(1-c)}(1-\chi)}{\chi(1-\chi_{(1-c)})} = r^*_{(1-c)}
\end{aligned}$$

And for the second part,

$$t_c + t_{(1-c)} = \frac{p_0\chi_0 - p_0\chi_0\chi_1 + p_1\chi_1 - p_1\chi_0\chi_1}{(\chi - \chi_0\chi_1)} = 1$$

□

Lemma 45. *If $\left|\tilde{t}_c - t_c\right| \leq 9\lambda^{1/3}$, $\left|\tilde{h}_c - h_c\right| \leq 3\lambda^{1/3}$ and $\tilde{r}_c = \min\{\tilde{t}_c, \max\{0, \tilde{h}_c - 3\lambda^{1/3}\}\}$, then $|\tilde{r}_c - t_c| \leq 9\lambda^{1/3}$ and $\tilde{r}_c \leq h_c$.*

Proof. Let $\tilde{a} = \max\{0, \tilde{h}_c - 3\lambda^{1/3}\}$. It is trivial to observe that $\tilde{a} \leq h_c$. We will show that $|\tilde{a} - h_c| \leq 6\lambda^{1/3}$, i.e. $\tilde{a}$ is a good approximation of h_c. If $\tilde{h}_c \geq 3\lambda^{1/3}$ then the result is trivial. Otherwise, $h_c \leq 6\lambda^{1/3}$ and, hence, $|\tilde{a} - p_c| \leq 6\lambda^{1/3}$.

If $\tilde{t}_c \leq \tilde{a}$ then $|\tilde{r}_c - t_c| \leq 9\lambda^{1/3}$. Otherwise, i.e. $\tilde{t}_c > \tilde{a}$, we need to consider two cases. If $t_c \leq \tilde{a}$ then $|\tilde{a} - t_c| = \tilde{a} - t_c \leq \tilde{t}_c - t_c = \left|\tilde{t}_c - t_c\right| \leq 9\lambda^{1/3}$. If $h_c \geq t_c \geq \tilde{a}$ (Lemma 44) then $|\tilde{a} - t_c| \leq |\tilde{a} - h_c| \leq 6\lambda^{1/3}$. Hence, $|\tilde{r}_c - t_c| \leq \max\{6\lambda^{1/3}, 9\lambda^{1/3}\} = 9\lambda^{1/3}$.

Moreover, $\tilde{r}_c \leq \tilde{a} \leq h_c$. □

Appendix C

Separations

C.1 Technical Results

We will talk about some technical results useful in Chapter 6. We state the following lemma without proof:

Lemma 46. *There exists a constant $c > 0$ such that for all $x \in [0, c]$, the following inequalities hold:*

$$(1 - x + x^2) \geq \exp(-x) \geq (1 - x) \geq \exp(-x - x^2)$$

Lemma 47. *Let $f(x, y) = 1 - (1 - {}^{1}/_{(N^3 - y)})^{N-x}$. For $\delta, \lambda = o(N)$, we have ${}^{(f(0,0) - f(\delta,\lambda))}/_{f(0,0)} \leq {}^{(\delta + o(1))}/_{(N - o(1))}$.*

Proof.

$$\begin{aligned}
\text{Denominator: } f(0,0) &= 1 - \left(1 - \frac{1}{N^3}\right)^N \\
&\geq 1 - \exp\left(-\frac{1}{N^2}\right) \\
&\geq 1 - \left(1 - \frac{1}{N^2} + \frac{1}{N^4}\right) = \frac{1}{N^2}\left(1 - \frac{1}{N^2}\right) \\
\text{Numerator: } f(0,0) - f(\delta, \lambda) &= \left(1 - \frac{1}{N^3 - \lambda}\right)^{N-\delta} - \left(1 - \frac{1}{N^3}\right)^N \\
&\leq \exp\left(-\frac{N - \delta}{N^3 - \lambda}\right) - \exp\left(-\frac{1}{N^2} - \frac{1}{N^5}\right) \\
&\leq 1 - \frac{N - \delta}{N^3 - \lambda} + \left(\frac{N - \delta}{N^3 - \lambda}\right)^2 - 1 + \frac{1}{N^2} + \frac{1}{N^5} \\
&= \frac{\delta}{N^3 - \lambda} - \frac{\lambda}{N^3(N^2 - o(1))} + \frac{1}{N^3}\left(\frac{N - \delta}{N^{3/2} - o(1)}\right)^2 + \frac{1}{N^5} \\
&\leq \frac{\delta + o(1)}{N^3 - \lambda} \\
\text{Therefore, } \frac{f(0,0) - f(\delta, \lambda)}{f(0,0)} &< \frac{\delta + o(1)}{N - o(1)}
\end{aligned}$$

□

References

[Bar86] David A. Mix Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc^1. In *STOC*, pages 1–5. ACM, 1986. 27

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *FOCS*, pages 106–115, 2001. 29, 105

[BCGL89] Shai Ben-David, Benny Chor, Oded Goldreich, and Michael Luby. On the theory of average case complexity. In *STOC*, pages 204–216. ACM, 1989. 8

[BCR86] Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. Information theoretic reductions among disclosure problems. In *FOCS*, pages 168–173. IEEE, 1986. 26

[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. 4, 14, 18, 19, 22, 24, 26, 33, 41, 44, 61, 104

[Bea95] Donald Beaver. Precomputing oblivious transfer. In Don Coppersmith, editor, *CRYPTO*, volume 963 of *Lecture Notes in Computer Science*, pages 97–109. Springer, 1995. 40

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC*, pages 103–112. ACM, 1988. 40, 75

[BGP00] Mihir Bellare, Oded Goldreich, and Erez Petrank. Uniform generation of np-witnesses using an np-oracle. *Inf. Comput.*, 163(2):510–526, 2000. 13, 61, 81, 82, 83

[BGS75] Theodore P. Baker, John Gill, and Robert Solovay. Relativizatons of the p =? np question. *SIAM J. Comput.*, 4(4):431–442, 1975. 3, 8

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *STOC*, pages 1–10. ACM, 1988. 4, 23, 24, 25, 28

[Bla79] G. R. Blakley. Safeguarding cryptographic keys. *Managing Requirements Knowledge, International Workshop on*, 0:313, 1979. 28

[Blu82] Manuel Blum. Coin flipping by telephone - a protocol for solving impossible problems. In *COMPCON*, pages 133–137. IEEE Computer Society, 1982. ix, 15, 16, 75, 76

[BM84] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM J. Comput.*, 13(4):850–864, 1984. 30

[BM88] Mihir Bellare and Silvio Micali. How to sign given any trapdoor function (extended abstract). In *STOC*, pages 32–42. ACM, 1988. 1

[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. 19, 20, 21, 29, 105, 106, 107, 109, 111, 113, 114, 115, 122

[BMG07] Boaz Barak and Mohammad Mahmoody-Ghidary. Lower bounds on signatures from symmetric primitives. In *FOCS*, pages 680–688. IEEE Computer Society, 2007. 105

[BMM99] Amos Beimel, Tal Malkin, and Silvio Micali. The all-or-nothing nature of two-party secure computation. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 80–97. Springer, 1999. 4, 27

[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. 30, 105

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*, pages 136–145, 2001. 6, 13, 25, 27, 31

[CCD88] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In *STOC*, pages 11–19. ACM, 1988. 4, 23, 24, 25, 28

[CF01] Ran Canetti and Marc Fischlin. Universally composable commitments. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 19–40. Springer, 2001. 11, 25

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *STOC*, pages 209–218, 1998. 106

[CGK90] Benny Chor, Mihály Geréb-Graus, and Eyal Kushilevitz. Private computations over the integers (extended abstract). In *FOCS*, volume I, pages 335–344. IEEE, 1990. 24

[CGK94] Benny Chor, Mihály Geréb-Graus, and Eyal Kushilevitz. On the structure of the privacy hierarchy. *J. Cryptology*, 7(1):53–60, 1994. 24, 29

[CGK95] Benny Chor, Mihály Geréb-Graus, and Eyal Kushilevitz. Private computations over the integers. *SIAM J. Comput.*, 24(2):376–386, 1995. 24

[CGMA85] Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *FOCS*, pages 383–395. IEEE, 1985. 28

[Cha82] David Chaum. Blind signatures for untraceable payments. In *CRYPTO*, pages 199–203, 1982. 30

[CI93] Richard Cleve and Russell Impagliazzo. Martingales, collective coin flipping and discrete control processes (extended abstract), 1993. 17, 18, 41, 44, 76, 77

[CI01] Benny Chor and Yuval Ishai. On privacy and partition arguments. *Inf. Comput.*, 167(1):2–9, 2001. 24, 29

[CK89] Benny Chor and Eyal Kushilevitz. A zero-one law for boolean privacy (extended abstract). In *STOC*, pages 62–72. ACM, 1989. 4, 24, 26, 34, 44, 104

[CK09] André Chailloux and Iordanis Kerenidis. Optimal quantum strong coin flipping. In *FOCS*, pages 527–533. IEEE Computer Society, 2009. 79, 127

[CKL03] Ran Canetti, Eyal Kushilevitz, and Yehuda Lindell. On the limitations of universally composable two-party computation without set-up assumptions. In Eli Biham, editor, *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 68–86. Springer, 2003. 4, 11, 23, 25, 34

[CKOR97] Ran Canetti, Eyal Kushilevitz, Rafail Ostrovsky, and Adi Rosén. Randomness vs. fault-tolerance. In *PODC*, pages 35–44, 1997. 24

[Cle86] Richard Cleve. Limits on the security of coin flips when half the processors are faulty (extended abstract). In *STOC*, pages 364–369. ACM, 1986. 41, 107

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *STOC*, pages 494–503, 2002. 4, 27, 40

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *STOC*, pages 151–158. ACM, 1971. 29

[CPS08] Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The random oracle model and the ideal cipher model are equivalent. In David Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2008. 29, 119

[Cré87] Claude Crépeau. Equivalence between two flavours of oblivious transfers. In Carl Pomerance, editor, *CRYPTO*, volume 293 of *Lecture Notes in Computer Science*, pages 350–354. Springer, 1987. 26

[CS95] Benny Chor and Netta Shani. The privacy of dense symmetric functions. *Computational Complexity*, 5(1):43–59, 1995. 24

[Dam87] Ivan Damgård. Collision free hash functions and public key signature schemes. In *EUROCRYPT*, pages 203–216, 1987. 30

[DG03] Ivan Damgård and Jens Groth. Non-interactive and reusable non-malleable commitment schemes. In *STOC*, pages 426–437. ACM, 2003. 61

[DKS99] Ivan Damgård, Joe Kilian, and Louis Salvail. On the (im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions. In *EUROCRYPT*, pages 56–73, 1999. 134

[DLMM11] Dana Dachman-Soled, Yehuda Lindell, Mohammad Mahmoody, and Tal Malkin. On the black-box complexity of optimally-fair coin tossing. In Yuval Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 450–467. Springer, 2011. 19, 20, 21, 106, 109

[Dol82] Danny Dolev. The byzantine generals strike again. *J. Algorithms*, 3(1):14–30, 1982. 25

[EGL85] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985. 26

[FGMO05] Matthias Fitzi, Juan A. Garay, Ueli M. Maurer, and Rafail Ostrovsky. Minimal complete primitives for secure multi-party computation. *J. Cryptology*, 18(1):37–61, 2005. 5, 25

[FM00] Matthias Fitzi and Ueli M. Maurer. From partial consistency to global broadcast. In *STOC*, pages 494–503, 2000. 5, 25

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986. 106

[GGK03] Rosario Gennaro, Yael Gertner, and Jonathan Katz. Lower bounds on the efficiency of encryption and digital signature schemes. In *STOC*, pages 417–425. ACM, 2003. 30

[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. 30, 105

[GIM$^+$10] S. Dov Gordon, Yuval Ishai, Tal Moran, Rafail Ostrovsky, and Amit Sahai. On complete primitives for fairness. In Daniele Micciancio, editor, *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 91–108. Springer, 2010. 5

[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, 2000. 1, 2, 29, 30, 39, 108, 124

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32. ACM, 1989. 16, 30, 76

[GMM07] Yael Gertner, Tal Malkin, and Steven Myers. Towards a separation of semantic and cca security for public key encryption. In Salil P. Vadhan, editor, *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 434–455. Springer, 2007. 30

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *STOC*, pages 291–304. ACM, 1985. 27, 28

[GMR01] Yael Gertner, Tal Malkin, and Omer Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *FOCS*, pages 126–135, 2001. 30, 105

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to prove all np-statements in zero-knowledge, and a methodology of cryptographic protocol design. In Andrew M. Odlyzko, editor, *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 171–185. Springer, 1986. 9, 27, 28

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *STOC*, pages 218–229. ACM, 1987. 4, 5, 26, 27, 29, 75

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in np have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991. 105

[Gol04] O. Goldreich. *Foundations of Cryptography: Basic applications*. Foundations of Cryptography. Cambridge University Press, 2004. 27

[GT00] Rosario Gennaro and Luca Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *FOCS*, pages 305–313, 2000. 30

[GV87] Oded Goldreich and Ronen Vainish. How to solve any protocol problem - an efficiency improvement. In Carl Pomerance, editor, *CRYPTO*, volume 293 of *Lecture Notes in Computer Science*, pages 73–86. Springer, 1987. 26

[Hai08] Iftach Haitner. Semi-honest to malicious oblivious transfer - the black-box way. In Ran Canetti, editor, *TCC*, volume 4948 of *Lecture Notes in Computer Science*, pages 412–426. Springer, 2008. 105

[Hås90] Johan Håstad. Pseudo-random generators under uniform assumptions. In *STOC*, pages 395–404. ACM, 1990. 2, 9, 16, 28

[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. 105

[HI08] Edward A. Hirsch and Dmitry Itsykson. An infinitely-often one-way function based on an average-case assumption. In Wilfrid Hodges and Ruy J. G. B. de Queiroz, editors, *WoLLIC*, volume 5110 of *Lecture Notes in Computer Science*, pages 208–217. Springer, 2008. 81

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999. 2, 9, 16, 28

[HKT11] Thomas Holenstein, Robin Künzler, and Stefano Tessaro. Equivalence of the random oracle model and the ideal cipher model, revisited. In *STOC*, 2011. 29, 119

[HM97] Martin Hirt and Ueli M. Maurer. Complete characterization of adversaries tolerable in secure multi-party computation (extended abstract). In *PODC*, pages 25–34, 1997. 25

[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. 105

[HO11] Iftach Haitner and Eran Omri. Coin flipping with constant bias implies one-way functions. To appear in FOCS, 2011. 79, 127

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *FOCS*, pages 230–235. IEEE, 1989. 1, 2, 8, 28, 29, 41, 63, 76, 77, 104, 105

[ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In *STOC*, pages 12–24. ACM, 1989. 2, 9, 13, 28, 61

[Imp95] Russell Impagliazzo. A personal view of average-case complexity. In *Structure in Complexity Theory Conference*, pages 134–147, 1995. 5, 8

[Imp09] Russell Impagliazzo. 5 worlds of problems. Talk at the workshop Complexity and Cryptography: Status of Impagliazzo's Worlds, Princeton, NJ. Video available from `http://intractability.princeton.edu/videos/stream/videoplay.html?videofile=cs/IW2009-500kb/Russel%20Impagliazzo.mp4`, 2009. 17, 76, 77

[Imp10] Russell Impagliazzo. Personal communication, 2010. 17, 76, 77

[Imp11] Russell Impagliazzo. Relativized separations of worst-case and average-case complexities for NP. In *IEEE Conference on Computational Complexity, San Jose, California, June 8-10*, 2011. 8

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *STOC*, pages 44–61. ACM, 1989. 1, 9, 19, 29, 30, 105, 106, 118

[IY87] Russell Impagliazzo and Moti Yung. Direct minimum-knowledge computations. In Carl Pomerance, editor, *CRYPTO*, volume 293 of *Lecture Notes in Computer Science*, pages 40–51. Springer, 1987. 28

[JVV86] Mark Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comput. Sci.*, 43:169–188, 1986. 13, 61, 81, 82

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In *STOC*, pages 20–31. ACM, 1988. 4, 26

[Kil91] Joe Kilian. A general completeness theorem for two-party games. In *STOC*, pages 553–560. ACM, 1991. 4, 18, 26, 33, 104

[Kil00] Joe Kilian. More general completeness theorems for secure two-party computation. In *STOC*, pages 316–324, 2000. 4, 27, 42, 45, 47, 48, 125

[Kit03] Alexei Kitaev. Quantum coin-flipping. Presentation at the 6th workshop on quantum information processing (qip 2003), 2003. 79

[KK05] Jonathan Katz and Chiu-Yuen Koo. On constructing universal one-way hash functions from arbitrary one-way functions. Cryptology ePrint Archive, Report 2005/328, 2005. `http://eprint.iacr.org/`. 9, 28

[KKMO00] Joe Kilian, Eyal Kushilevitz, Silvio Micali, and Rafail Ostrovsky. Reducibility and completeness in private computations. *SIAM J. Comput.*, 29(4):1189–1208, 2000. 26

[KM11] Daniel Kraschewski and Jörn Müller-Quade. Completeness theorems with constructive proofs for finite deterministic 2-party functions. In Yuval Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 364–381. Springer, 2011. 23, 27, 65

[KMO94] Eyal Kushilevitz, Silvio Micali, and Rafail Ostrovsky. Reducibility and completeness in multi-party private computations. In *FOCS*, pages 478–489. IEEE, 1994. 4, 6, 18, 26

[KMR09] Robin Künzler, Jörn Müller-Quade, and Dominik Raub. Secure computability of functions in the it setting with dishonest majority and applications to long-term security. In Omer Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 238–255. Springer, 2009. 4, 18, 19, 22, 24, 26, 44, 49

[KN04] Iordanis Kerenidis and Ashwin Nayak. Weak coin flipping with small bias. *Inf. Process. Lett.*, 89(3):131–135, 2004. 76

[KOR96] Eyal Kushilevitz, Rafail Ostrovsky, and Adi Rosén. Characterizing linear size circuits in terms of privacy. In *STOC*, pages 541–550, 1996. 24

[Kre11] Gunnar Kreitz. A zero-one law for secure multi-party computation with ternary outputs. In Yuval Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 382–399. Springer, 2011. 4, 18, 24, 26, 27, 34, 104

[KSS00] Jeff Kahn, Michael E. Saks, and Clifford D. Smyth. A dual version of reimer's inequality and a proof of rudich's conjecture. In *IEEE Conference on Computational Complexity*, pages 98–103, 2000. 30

[KSS11] Jeff Kahn, Michael Saks, and Clifford D. Smyth. The dual bkr inequality and rudich's conjecture. *Combinatorics, Probability & Computing*, 20(2):257–266, 2011. 30

[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. 105

[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. 30, 105

[Kus89] Eyal Kushilevitz. Privacy and communication complexity. In *FOCS*, pages 416–421. IEEE, 1989. 4, 11, 14, 18, 19, 22, 24, 26, 33, 41, 44, 61, 104

[Lev85] Leonid A. Levin. One-way functions and pseudorandom generators. In *STOC*, pages 363–365. ACM, 1985. 1

[Lin04] Yehuda Lindell. Lower bounds for concurrent self composition. In Moni Naor, editor, *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 203–222. Springer, 2004. 4, 11, 23, 25, 34

[LP09] Yehuda Lindell and Benny Pinkas. A proof of security of yao's protocol for two-party computation. *J. Cryptology*, 22(2):161–188, 2009. 27

[LSP82] Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982. 25

[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. 105

[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. 30, 105

[MMP11] Mohammad Mahmoody, Hemanta K. Maji, and Manoj Prabhakaran. Limits of random oracles in secure computation. Manuscript, 2011. iii, 3, 10, 19, 105, 106, 108, 109, 120, 124, 126

[MNS09] Tal Moran, Moni Naor, and Gil Segev. An optimally fair coin toss. In Omer Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2009. 17, 41, 77

[Moc07] Carlos Mochon. Quantum weak coin flipping with arbitrarily small bias, 2007. 79

[MOPR11] Hemanta K. Maji, Pichayoot Ouppaphan, Manoj Prabhakaran, and Mike Rosulek. Exploring the limits of common coins using frontier analysis of protocols. In Yuval Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2011. iii, 3, 10, 11, 12, 23, 125

[MP11] Hemanta K. Maji and Manoj Prabhakaran. Limits of common coins: Further results. Submitted to Indocrypt, 2011. iii, 3, 10, 125

[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`. 3, 4, 10, 11, 14, 15, 18, 19, 22, 24, 26, 32, 41, 42, 44, 49, 53, 54, 60, 61, 62, 63, 64, 65, 104, 106, 107, 109

[MPR10a] 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. iii, 3, 10, 40, 41, 42, 69

[MPR10b] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. A zero-one law for cryptographic complexity with respect to computational uc security. In Tal Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 595–612. Springer, 2010. 3, 4, 5, 13, 14, 15, 27, 60, 61, 63, 65, 66

[MPS10] Hemanta K. Maji, Manoj Prabhakaran, and Amit Sahai. On the computational complexity of coin flipping. In *FOCS*, pages 613–622. IEEE Computer Society, 2010. iii, 3, 10, 17, 127

[Nao89] Moni Naor. Bit commitment using pseudo-randomness. In Gilles Brassard, editor, *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 128–136. Springer, 1989. 9, 16, 28

[NW88] Noam Nisan and Avi Wigderson. Hardness vs. randomness (extended abstract). In *FOCS*, pages 2–11. IEEE, 1988. 9

[NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *STOC*, pages 33–43. ACM, 1989. 1, 2, 9, 28, 30

[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. 13, 28, 61, 105

[OW93] Rafail Ostrovsky and Avi Wigderson. One-way fuctions are essential for non-trivial zero-knowledge. In *ISTCS*, pages 3–17, 1993. 13, 28, 61, 81, 103, 105

[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. 4, 11, 14, 25, 34, 45, 61, 66

[Rab81] Michael O. Rabin. How to exchange secrets with oblivious transfer. Technical Report TR-81, Aiken Computation Lab, Harvard University, 1981. 26

[RB89] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *STOC*, pages 73–85. ACM, 1989. 4, 25

[Rog91] Phillip Rogaway. *The Round Complexity of Secure Protocols.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1991. 27

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC*, pages 387–394. ACM, 1990. 1, 2, 9, 28

[Ros09] Mike Rosulek. *The Structure of Secure Multi-Party Computation.* PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 2009. 3, 10

[Ros11] Mike Rosulek. Universal composability from essentially any trusted setup. Cryptology ePrint Archive, Report 2011/240, 2011. `http://eprint.iacr.org/`. 13

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In Omer Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 419–436. Springer, 2009. 30

[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. 29, 105

[Rud88] Steven Rudich. *Limits on the Provable Consequences of One-way Functions.* PhD thesis, University of California at Berkeley, 1988. 30

[Rud91] Steven Rudich. The use of interaction in public cryptosystems (extended abstract). In Joan Feigenbaum, editor, *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 242–251. Springer, 1991. 9, 30

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979. 28

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984. 30

[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. 30, 105

[SY90] Alfredo De Santis and Moti Yung. On the design of provably secure cryptographic hash functions. In *EUROCRYPT*, pages 412–431, 1990. 28

[Uni95] United States Department of Commerce. Secure hash standard. Technical Report FIPS PUB 180-1, US Department of Commerce, April 1995. 1

[Vah10] Yevgeniy Vahlis. Two is a crowd? a black-box separation of one-wayness and security under correlated inputs. In Daniele Micciancio, editor, *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 165–182. Springer, 2010. 30

[Wie83] Stephen Wiesner. Conjugate coding. *Sigact News*, 15:78–88, 1983. 26

[Yao82a] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *FOCS*, pages 160–164. IEEE, 1982. 22, 32

[Yao82b] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *FOCS*, pages 80–91. IEEE, 1982. 28

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FOCS*, pages 162–167. IEEE, 1986. 4, 27, 29

[Zac86] Stathis Zachos. Probabilistic quantifiers, adversaries, and complexity classes: An overview. In Alan L. Selman, editor, *Structure in Complexity Theory Conference*, volume 223 of *Lecture Notes in Computer Science*, pages 383–400. Springer, 1986. 17